ABSTRACT

Title of Dissertation: ON ALGORITHMS, FAIRNESS, AND INCENTIVES

Seyed A. Esmaeili
Doctor of Philosophy, 2023

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

Much of decision making is now rendered at least partly through algorithms which were originally designed to optimize an objective such as accuracy or revenue while mostly ignoring the possible unfairness or harm that could be caused. As a result several case studies have demonstrated empirically that deployed algorithmic decision making systems do in fact violate standard notions of fairness. This has made the issue of fairness an important consideration in algorithm design. In this thesis we will consider fair variants of fundamental and important problems in machine learning and operations research.

We start by considering clustering where we focus on a common group (demographic) fairness notion and address important variants of it: (I) We start with the frequent case where group memberships are imperfectly known. Based on stochastic optimization we propose probabilistic fair clustering which is a generalization of fair clustering that handles the case of unknown group memberships. We derive approximation algorithms for this setting and empirically test their performance. (II) As largely known in fair algorithms achieving fairness mostly comes at the

expense of degrading the value of the optimization objective. In fact, in the case of fair clustering the degradation (price of fairness) can be unbounded. To handle this, we propose fair clustering under a bounded cost where we define a measure of unfairness and minimize this measure subject to a pre-set upper bound on the clustering objective. We derive lower bounds on the approximation ratio and give approximation algorithms as well. (III) We consider the downstream effects of clustering where the center (prototype) of each cluster is examined and each cluster is assigned a label of a specific quality based on its prototype. These labels are shared over a collection of clusters and as a result traditional fair clustering is too restrictive. We therefore propose fair labeled clustering where proportional demographic representation is to be preserved over the labels instead of the clusters and derive algorithms for it. (IV) Motivated by the fact that at least seven different fairness notions have been introduced so far in fair clustering, we take a step towards understanding how these notions relate to one another. Specifically, we consider two group representation-based fairness notions and show that an approximation algorithm for one can be used to satisfy both notions simultaneously at a bounded degradation to the approximation ratio. We further show how these two notions are incompatible with a collection of distance-based fairness notions in clustering.

We then move to another problem, namely online bipartite matching which in its most common form involves three interacting entities: two sides (buyers and sellers) to be matched and the platform operator. Unlike the existing literature we derive online algorithms with competitive ratio guarantees for the operator's revenue as well as fairness guarantees for the two sides to be matched, thus providing utility guarantees for all sides of the market.

Finally, we consider a problem where the incentives of the individuals and organizations involved is a major consideration. Specifically, we consider the problem of redistricting and

gerrymandering. Inspired by the Kemeny rule for rank aggregation, we introduce a simple and interpretable family of distances over redistricting maps and define the medoid map which mirrors the Kemeny ranking. Interestingly, we show that a by-product of our framework is that it can detect some gerrymandered instances. Specifically, the 2011 and 2016 enacted maps of North Carolina and the 2011 enacted map of Pennsylvania (all considered to be gerrymandered) are all shown to be at least in the 99th distance percentile in comparison to an ensemble of redistricting maps. This gives a significant advantage in gerrymandering detection since the previous methods relied on election outcomes whereas our method is purely distance-based.

ON ALGORITHMS, FAIRNESS, AND INCENTIVES

by

Seyed Adulaziz Esmaeili

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
2023

Advisory Committee:
Associate Professor John P. Dickerson, Chair/Advisor
Professor Aravind Srinivasan, Co-Advisor
Professor S. Raghavan, Dean's Representative
Dr. Aleksandrs Slivkins
Assistant Professor Laxman Dhulipala

© Copyright by
Seyed Abdulaziz Esmaeili
2023

Dedication

To my family.

Acknowledgments

I would like to express my sincere gratitude to my academic advisors, John P. Dickerson and Aravind Srinivasan. Their insights and help in various projects and general academic guidance has made this milestone possible and much easier. I would also like to thank them for being available for meetings and enabling many career opportunities.

I would also like to thank my collaborators for their valuable insights and contributions: Brian Brubach, Darshan Chakrabarti, Davidson Cheng, Sharmila Duppala, Haley Grape, Christine Herlihy, Marina Knittel, Jamie Morgenstern, Vedant Nanda, Suho Shin, Alex Slivkins, Daniel Smolyak, Leonidas Tsepenekas, and Claire Zhang.

I would like to thank S. Raghu Raghavan, Alex Slivkins, and Laxman Dhulipala for taking the time to serve on my defense committee. In addition, I would like to thank Ian Miers for serving on my proposal committee.

Tom Hurst has provided me with so much help over the years that have made things much easier and smoother and I'm very grateful for his assistance. I would like to thank my friends Shaopeng Zhu and Xuchen You for their advice and encouragement.

Finally, I would like to thank my parents for giving me the upbringing that made this achievement possible. I am also very grateful for my sister and brothers, their positive influence over me has been and continues to be immense. I dedicate this thesis to them.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Fairness has become an important consideration in algorithm design. An unsurprising fact given the widespread use of algorithms in decision making systems that have serious consequence on the lives of individuals. Recidivism prediction [2], kidney exchange [3], loan approval [4], hiring [5], and many others are examples of real-life situations where algorithms now play a key role in deciding the outcome. Several studies have documented what can be easily considered fairness violations done by algorithmic decision making systems, see for example [6, 7] for many instances of algorithmic fairness violations.

Given the undeniable fact that algorithms can indeed violate various notions of fairness, the recent years have seen a significant surge in the design of fair algorithms. In this thesis we address fairness issues in a collection of fundamental problems in machine learning and operations research. We start with clustering –arguably the most fundamental problem in unsupervised learning– and consider various notions of group fairness. We then move to online bipartite matching where in contrast to previous work we consider the welfare of all three sides of the market: the platform operator and the two sides to be matched.

In addition to fairness, the incentives of individuals and organizations is also an important consideration which algorithm design should take into account. In this direction, we consider redistricting –a problem which dates back to over two centuries– where inspired by the Kemeny

rule we introduce a simple and interpretable family of distances over redistricting maps and define a medoid map which mirrors the Kemeny ranking. Interestingly, we show that a by product of our framework is that it can detect gerrymandered instances. Specifically, the 2011 and 2016 enacted maps of North Carolina and the 2011 enacted map of Pennsylvania (all considered to be gerrymandered) are all shown to be at least in the 99^{th} distance percentile in comparison to an ensemble of redistricting maps.

1.1 Overview of the Thesis

Chapter 2 is dedicated to fair clustering where a collection of group fairness notions are studied. Chapter 3 is concerned with fairness in online matching. Finally, in chapter 4 we discuss our work on redistricting and gerrymandering. Finally, in chapter 5 we point out some remarks and possible opportunities for future work specifically for fair clustering and redistricting/gerrymandering. Below, we give a more detailed description of the results of chapters 2, 3, and 4.

1.1.1 Fair Clustering

The fair clustering problem was introduced by [8]. In its most basic form the problem receives as input a collection of points in a metric space that are to be grouped into k clusters according to some clustering objective such as the k-means. However, unlike the typical setting each point has a color associated with it which indicates its group membership. Based on disparate impact [9] the clustering is fair if each cluster contains close to population level proportions of each color. We identify various modifications and generalizations of this setting and

give algorithms with theoretical guarantees for them. We consider a collection of problems in this setting as discussed below.

1.1.1.1 Probabilistic Fair Clustering

In many applications group memberships may not be fully known (see [10–13]). This can be caused by various factors such as the the group memberships not being available in the dataset, incorrect or noisy reporting of group memberships. This issue requires a generalization of fair clustering. We therefore introduce probabilistic fair clustering where the group memberships are known probabilistically instead of deterministically. More specifically, in the typical fair clustering setting each point j has a single color associated with it from the set of colors $\mathcal{H}$. However, in probabilistic fair clustering, each point j now has a set of values p_j^h associated with it where p_j^h denotes the probability that point j has color h, clearly $p_j^h \geq 0$ and $\sum_{h \in \mathcal{H}} p_j^h = 1$. Having generalized the color assignments to the probabilistic setting, we also generalize the fairness constraint. Specifically, now the fairness constraint is for each cluster to have the right proportions of colors in expectation instead of deterministically. This setting proves to be much more challenging then the deterministic setting. For the two color case, we show algorithms with bounded fairness violations. For the multiple color case, we show a fixed parameter tractable algorithm under an assumption that the size of each cluster in the optimal solution is lower bounded by some large enough value.

This work was published in NeurIPS-2020, see [14].

1.1.1.2 Fair Clustering under a Bounded Cost

Like most optimization problem once a fairness constraint is imposed, a degradation in the optimization objective is mostly unavoidable [15, 16]. In fact, in the case of fair clustering the degradation in the clustering objective, i.e. the price of fairness is unbounded. We therefore, introduce the problem of fair clustering under a bounded cost where instead of minimizing the clustering objective subject to the fairness constraint, we are instead given a pre-set upper bound on the clustering cost and we are supposed to minimize a measure of "unfairness" instead. Clearly, the first issue in this problem is to define a suitable measure of unfairness, this is done using standard notions from welfare economics. Specifically, we define three notions of unfairness objectives to be minimized based on the utilitarian, egalitarian, and leximin objectives. Having defined these objectives, we prove that these problems are NP-hard and derive lower bounds on the approximation guarantees of any algorithm. We then show algorithms with theoretical guarantees for this setting.

This work was published in NeurIPS-2021, see [17].

1.1.1.3 Fair Labeled Clustering

A lacking issue in fair clustering work and arguably in clustering in general [18] is that the downstream effects are not often considered. It is natural in a clustering setting to expect that different clusters will receive different outcomes and that these outcomes will be of a different quality. Therefore, an individual in a given cluster receives a specific utility for being in that cluster. Accordingly, it maybe desired to maintain the right demographic proportions not within

the clusters but rather across the labels of the clusters. We therefore introduce the problem of fair labeled clustering. In fair labeled clustering each cluster (center) has a label associated with it and we minimize the clustering objective subject to having the right proportions across the labels instead of within the clusters. We further consider other constraints on this problem such as lower and upper bounds on the number of individuals in each label. Moreover, we consider a setting where the labels of the centers are assigned and known and where the labels are unknown and free to be assigned subject to possible lower and upper bounds on the number of centers of each label.

This work was published in KDD-2022, see [19].

1.1.1.4 Doubly Constrained Fair Clustering

In addition to the group fairness constraint considered in the previous outlined research, there have been other fairness notions that had been considered in fair clustering. One can count six more different constraints that had been considered in the literature as well [20]. The issue is that while each constraint is well-justified, it is considered exclusively in isolation. Therefore, the relation between these constraints and how one may satisfy more than one constraint simultaneously is an important question. We take the first step in this direction. Specifically, we consider the above mentioned group fairness (**GF**) notion of [21] and the diversity in center selection (**DS**) notion of [22, 23] which essentially states that different groups should be represented in the selected centers according to some pre-set lower and upper bound values specific to each group, thereby ensuring group diversity in the selected centers. We show that given an approximation

algorithm for either problem (**GF** or **DS** only) the solution can be post-processed to satisfy both constraints simultaneously at a bounded degradation to the clustering cost. We also study the price of fairness –the degradation in the clustering cost due to imposing fairness constraints– and show that any **GF** solution can be post-processed at a bounded degradation to the clustering cost whereas the reverse is not true. Further, we show the **GF** and **DS** are each incompatible with a collection of distance based fairness notions, i.e. having an empty feasible set in general.

This work is currently under review, see [24].

1.1.2 Fairness in Online Bipartite Matching

Matching is among the most fundamental problems in combinatorial optimization and economics. The online and bipartite variant of matching [25, 26] has found numerous applications in many domains including ad allocation, ridesharing, and crowdsourcing to name a few. Since the vertices on one side of the bipartite graph are often used to represent users, workers, or drivers and on the other side they represent ads, employers, or riders; the matching outcomes and their quality have a clear effect on the welfare of the vertices (which are workers or employers, etc). Therefore, fairness considerations in this setting are well-founded. In most online matching applications, there are three entities: the platform operator and two sides to be matched. The previous work has considered fairness/utility in this setting [27–29] but only for one or two sides simultaneously. In our work, we consider Rawlsian fairness considerations for each side as well as utility considerations for the platform operator. We consider both individual and group fairness. Moreover, we show hardness results which bound the performance of any algorithm as well

as other results that show an intrinsic conflict between individual and group notions of fairness.

This work was published in AAAI 2023, see [30].

1.1.3 Implications of Distance over Redistricting Maps: Central and Outlier Maps

Redistricting is an important problem in any representative democracy. Given a state, redistricting is the process of partitioning the state into a collection of districts such that a collection of rules are satisfied. The most common of these rules are contiguity, equal population, and compactness. Further additional rules are also often imposed and some are state-dependent. The ambiguity of these rules, leads to a large collection of possible redistricting maps and therefore the entity in charge of redistricting (often a collection of elected representatives) tend to select maps that place their political party at an unfair advantage. This phenomenon is well-recognized and referred to as gerrymandering [31]. As a result, in the last decade a collection of Markov Chain Monte Carlo (MCMC) methods for sampling valid redistricting maps were introduced [32, 33]. These MCMC based methods could in some situations be used to show that an enacted map is gerrymandered. For example, a histogram over the set of all possible redistricting maps of the won seats for each party can be estimated. Using such a histogram, it maybe possible to arrive at the conclusion that the election outcome is "rare". That is, it occurs only over a small set of maps in comparison to the whole set of other possible maps. Arguments of this kind were used to show that some enacted redistricting maps are in fact gerrymandered [34].

In our work we study the implications of a distance measure over redistricting maps. More

specifically, inspired by the Kemeny rule, we introduce a distance measure over redistricting maps and define the medoid map which mirrors the Kemeny ranking. We discuss computational and statistical results for obtaining the medoid map. Furthermore, we also define the centroid map –which is not a valid map but a helpful mathematical object– and discuss some of its important properties and applications. Finally, we show that our framework can be used to detect gerrymandered instances. Specifically, the 2011 and 2016 enacted maps of North Carolina and the 2011 enacted map of Pennsylvania (all considered to be gerrymandered) are all shown to be at least in the 99^{th} distance percentile in comparison to an ensemble of redistricting maps. Importantly, unlike previous methods for detecting gerrymandered maps our distance based method does not use election results.

This work is currently under review, see [35].

Chapter 2: Fair Clustering

This chapter focuses on fair clustering. Most of the sections are dedicated to generalizations of the original group fairness definition that was introduced in [8]. Further, we focus on center based clustering objectives, i.e. k-center, k-median, and k-means clustering.

2.1 Preliminaries

Let $\mathcal{C}$ be the set of points in a metric space with distance function $d : \mathcal{C} \times \mathcal{C} \to \mathbf{R}_{\geq 0}$. The distance between a point j and a set S is defined as $d(j, S) = \min_{j \in S} d(j, j)$. In a k-clustering an objective function $L^k(\mathcal{C})$ is given, a set $S \subseteq \mathcal{C}$ of at most k points must be chosen as the set of centers, and each point in $\mathcal{C}$ must get assigned to a center in S through an assignment function $\phi : \mathcal{C} \to S$, forming a k-partition of the original set: $\mathcal{C}_1, \ldots, \mathcal{C}_k$. The optimal solution is defined as a set of centers and an assignment function that minimizes the objective $L^k(\mathcal{C})$. The well known k-center, k-median, and k-means can all be stated as the following problem:

$$\min_{S:|S|\leq k,\phi} L_p^k(\mathcal{C}) = \min_{S:|S|\leq k,\phi} \Big(\sum_{j \in \mathcal{C}} d^p(j, \phi(j)) \Big)^{1/p} \tag{2.1}$$

where p equals $\infty, 1,$ and 2 for the case of the k-center, k-median, and k-means, respectively. For such problems the optimal assignment for a point j is the nearest point in the chosen set of

centers S. However, in the presence of additional constraints such as imposing a lower bound on the cluster size [36] or an upper bound [37, 38] this property no longer holds. In general, this is also true in fair clustering.

To formulate the fair clustering problem, a set of colors $\mathcal{H} = \{h_1, \ldots, h, \ldots, h_m\}$ is introduced and each point j is mapped to a color through a given function $\chi : \mathcal{C} \to \mathcal{H}$. Previous work in fair clustering [1, 8, 39, 40] adds to (2.1) the following proportional representation constraint, i.e.:

$$\forall i \in S, \forall\, h \in \mathcal{H} : l_h |\, \mathcal{C}_i \,| \leq |\, \mathcal{C}_{i,h} \,| \leq u_h |\, \mathcal{C}_i \,| \tag{2.2}$$

where $\mathcal{C}_{i,h_\ell}$ is the set of points in cluster i having color h_ℓ. The bounds $l_h, u_h \in (0, 1)$ are given lower and upper bounds on the desired proportion of a given color in each cluster, respectively.

Objective 2.1 and constraint 2.2 are essentially the main problem we study in this chapter. In particular, in section 2.2 we consider the problem under probabilistic knowledge of group memberships. Moreover, in section 2.3 we consider the problem is what is essentially a reverse format, i.e. the clustering cost is set as a constraint and an unfairness measure is instead minimized. Additionally, in section 2.4 we consider fairness while thinking about the downstream effects of clustering. Specifically, we consider the case where each center has a label assigned to it and where we are supposed to achieve demographic fairness over the labels not the centers. Finally, in section 2.5 we consider the constraint of 2.2 and another fair clustering constraint simultaneously. Specifically, we consider the diversity in center selection constraint [22, 23]. We show algorithms that satisfy both constraints simultaneously. Further, we show how both of these constraints are incompatible with a collection of distance based fair clustering notions.

2.2 Probabilistic Fair Clustering

In many applications the group memberships (colors of each point or vertex) are not fully known (see [10–13]). Therefore, we introduce probabilistic fair clustering where we generalize the problem by assuming that the color of each point is not known deterministically but rather probabilistically. That is, each point j has a given value p_j^h for each $h \in \mathcal{H}$, representing the probability that point j has color h, with $\sum_{h\in\mathcal{H}} p_j^h = 1$.

The constraints are then modified to have the expected color of each cluster fall within the given lower and upper bounds. This leads to the following optimization problem:

$$\min_{S:|S|\leq k,\phi} L_p^k(\mathcal{C}) \tag{2.3a}$$

$$\text{s.t. } \forall i \in S, \forall\, h \in \mathcal{H} : l_{h_\ell}|\phi^{-1}(i)| \leq \sum_{j\in\phi^{-1}(i)} p_j^{h_\ell} \leq u_{h_\ell}|\phi^{-1}(i)| \tag{2.3b}$$

Following [40], we define a γ violating solution to be one for which for all $i \in S$:

$$l_{h_\ell}|\phi^{-1}(i)| - \gamma \leq \sum_{j\in\phi^{-1}(i)} p_j^{h_\ell} \leq u_{h_\ell}|\phi^{-1}(i)| + \gamma \tag{2.4}$$

This effectively captures the amount γ, by which a solution violates the fairness constraints.

2.2.1 Approximation Algorithms and Theoretical Guarantees

We essentially have two algorithms although they involve similar steps. One algorithm is for the two-color case which is discussed in section (2.2.1.1) and the other algorithm is for the multiple-color case under a large cluster assumption which is discussed in section (2.2.1.2).

2.2.1.1 Algorithms for the Two Color Case

Our algorithm follows the two step method of [40], although we differ in the LP rounding scheme. Let $\mathrm{PFC}(k, p)$ denote the probabilistic fair clustering problem. The color-blind clustering problem, where we drop the fairness constraints, is denoted by $\mathrm{Cluster}(k, p)$. Further, define the fair assignment problem $\text{FA-PFC}(S, p)$ as the problem where we are given a fixed set of centers S and the objective is to find an assignment ϕ minimizing $L_p^k(\mathcal{C})$ and satisfying the fairness constraints 2.3b for probabilistic fair clustering. We prove the following (similar to theorem 2 in [40]):

Theorem 2.2.1. *Given an α-approximation algorithm for $\mathrm{Cluster}(k, p)$ and a γ-violating algorithm for $\text{FA-PFC}(S, p)$, a solution with approximation ratio $\alpha + 2$ and constraint violation at most γ can be achieved for $\mathrm{PFC}(k, p)$.*

Proof. Let $\mathcal{I}_{\mathrm{PFC}}$ a given instance of $\mathrm{PFC}(k, p)$, $SOL_{\mathrm{PFC}} = (S^*_{\mathrm{PFC}}, \phi^*_{\mathrm{PFC}})$ is an optimal solution of $\mathcal{I}_{\mathrm{PFC}}$ and $\mathrm{OPT}_{\mathrm{PFC}}$ is its corresponding optimal value. Also, for $\mathrm{Cluster}(k, p)$ and for any instance of it, the optimal value is denoted by $\mathrm{OPT}_{\mathrm{Cluster}}$ and the corresponding solution by $\mathrm{SOL}_{\mathrm{Cluster}} = (S^*_{\mathrm{Cluster}}, \phi^*_{\mathrm{Cluster}})$.

The proof closely follows that from [40]. First running the color-blind α approximation algorithm results in a set of centers S, an assignment ϕ, and a solution value that is at most $\alpha\mathrm{OPT}_{\mathrm{Cluster}} \leq \alpha\,\mathrm{OPT}_{\mathrm{PFC}}$. Note that $\mathrm{OPT}_{\mathrm{Cluster}} \leq \mathrm{OPT}_{\mathrm{PFC}}$ since $\mathrm{PFC}(k, p)$ is a more constrained problem than $\mathrm{Cluster}(k, p)$. Now we establish the following lemma:

Lemma 2.2.2. $\mathrm{OPT}_{\text{FA-PFC}} \leq (\alpha + 2)\,\mathrm{OPT}_{\mathrm{PFC}}$

Proof. The lemma is established by finding the instance satisfying the inequality. Let $\phi'(j) =$

$\arg\min_{i \in S} d(i, \phi^*_{\text{PFC}}(j))$, i.e. an assignment that routes the vertices from the optimal center to the nearest center in color-blind solution S. For any point j the following holds:

$$\begin{aligned} d(j, \phi'(j)) &\leq d(j, \phi^*_{\text{PFC}}(j)) + d(\phi^*_{\text{PFC}}(j), \phi'(j)) \\ &\leq d(j, \phi^*_{\text{PFC}}(j)) + d(\phi^*_{\text{PFC}}(j), \phi(j)) \\ &\leq d(j, \phi^*_{\text{PFC}}(j)) + d(j, \phi^*_{\text{PFC}}(j)) + d(j, \phi(j)) \\ &= 2d(j, \phi^*_{\text{PFC}}(j)) + d(j, \phi(j)) \end{aligned}$$

stacking the distance values in the vectors $\vec{d}(j, \phi'(j))$, $\vec{d}(j, \phi^*_{\text{PFC}}(j))$, and $\vec{d}(j, \phi(j))$. By the virtue of the fact that $\big(\sum_{j \in \mathcal{C}} x^p(j)\big)^{1/p}$ is the ℓ_p-norm of the associated vector $\vec{x}$ and since each entry in $\vec{d}(j, \phi'(j))$ is non-negative, the triangular inequality for norms implies:

$$\begin{aligned} \big(\sum_{j \in \mathcal{C}} d^p(j, \phi'(j))\big)^{1/p} &\leq 2\big(\sum_{j \in \mathcal{C}} d^p(j, \phi^*_{\text{PFC}}(j))\big)^{1/p} \\ &+ \big(\sum_{j \in \mathcal{C}} d^p(j, \phi(j))\big)^{1/p} \end{aligned}$$

It remains to show that ϕ' satisfies the fairness constraints 2.3b, for any color h and any center i in S, denote $N(i) = \{j \in S^*_{\text{PFC}} | \arg\min_{i' \in S} d(i', j) = i\}$, then we have:

$$\frac{\sum_{j \in \phi'^{-1}(i)} p_j^h}{|\phi'^{-1}(i)|} = \frac{\sum_{j \in N(i)} \Big(\sum_{j \in \phi^{*-1}_{\text{PFC}}(j)} p_j^h\Big)}{\sum_{j \in N(i)} |\phi^{*-1}_{\text{PFC}}(j)|}$$

It follows by algebra and the lower and upper fairness constrain bounds satisfied by ϕ^*_{PFC}:

$$
\begin{aligned}
l_h &\leq \min_{j \in N(i)} \frac{\left(\sum_{j \in {\phi^*}^{-1}_{\mathrm{PFC}}(j)} p_j^h\right)}{|{\phi^*}^{-1}_{\mathrm{PFC}}(j)|} \\
&\leq \frac{\sum_{j \in N(i)} \left(\sum_{j \in {\phi^*}^{-1}_{\mathrm{PFC}}(j)} p_j^h\right)}{\sum_{j \in N(i)} |{\phi^*}^{-1}_{\mathrm{PFC}}(j)|} \\
&\leq \max_{j \in N(i)} \frac{\left(\sum_{j \in {\phi^*}^{-1}_{\mathrm{PFC}}(j)} p_j^h\right)}{|{\phi^*}^{-1}_{\mathrm{PFC}}(j)|} \\
&\leq u_h
\end{aligned}
$$

This shows that there exists an instance for FA-PFC that both satisfies the fairness constraints and has cost $\leq 2\,\mathrm{OPT}_{\mathrm{PFC}} + \alpha \mathrm{OPT}_{\mathrm{Cluster}} \leq (\alpha + 2)\,\mathrm{OPT}_{\mathrm{PFC}}$. □

Now combining the fact that we have an α approximation ratio for the color-blind problem, along with an algorithm that achieves a γ violation to FA-PFC with a value equal to the optimal value for FA-PFC, the proof for theorem 2.2.1 is complete. □

Now we describe the steps of the algorithm:

Step 1, Color-Blind Approximation Algorithm: At this step an ordinary (color-blind) α-approximation algorithm is used to find the cluster centers. For example, the Gonzalez algorithm [41] can be used for the k-center problem or the algorithm of [42] can be used for the k-median. This step results in a set S of cluster centers. Since this step does not take fairness into account, the resulting solution does not necessarily satisfy constraints 2.3b for probabilistic fair clustering.

Step 2, Fair Assignment Problem: In this step, a linear program (LP) is set up to satisfy the fairness constraints. The variables of the LP are x_{ij} denoting the assignment of point j to

center i in S. Specifically, the LP is:

$$\min \sum_{j \in \mathcal{C}, i \in S} d^p(i,j) x_{ij} \tag{2.5a}$$

$$\text{s.t. } \forall i \in S \text{ and } \forall\, h \in \mathcal{H} : \tag{2.5b}$$

$$l_h \sum_{j \in \mathcal{C}} x_{ij} \leq \sum_{j \in \mathcal{C}} p_j^h x_{ij} \leq u_h \sum_{j \in \mathcal{C}} x_{ij} \tag{2.5c}$$

$$\forall j \in \mathcal{C} : \sum_{i \in S} x_{ij} = 1, \quad 0 \leq x_{ij} \leq 1 \tag{2.5d}$$

Since the LP above is a relaxation of FA-PFC(S,p), we have $\mathrm{OPT}^{\mathrm{LP}}_{\text{FA-PFC}} \leq \mathrm{OPT}_{\text{FA-PFC}}$. We note that for k-center there is no objective, instead we have the following additional constraint: $x_{ij} = 0$ if $d(i,j) > w$ where w is a guess of the optimal radius. Also, for k-center the optimal value is always the distance between two points. Hence, through a binary search over the polynomially-sized set of distance choices we can obtain the minimum satisfying distance.

What remains is to round the fractional assignments x_{ij} resulting from solving the LP.

Step 3, Rounding for the Two Color Case: Our rounding method is based on calculating a minimum-cost flow in a carefully constructed graph. For each $i \in S$, a set C_i with $|C_i| = \left\lceil \sum_{j \in \mathcal{C}} x_{ij} \right\rceil$ vertices is created. Moreover, the set of vertices assigned to cluster i, i.e. $\phi^{-1}(i) = \{j \in \mathcal{C} \,|x_{ij} > 0\}$ are sorted in a non-increasing order according to the associated probability p_j and placed into the array $\vec{A}_i$. A vertex in C_i (except possibly the last) is connected to as many vertices in $\vec{A}_i$ by their sorting order until it accumulates an assignment value of 1. A vertex in $\vec{A}_i$ may be connected to more than one vertex in C_i if that causes the first vertex in C_i to accumulate an assignment value of 1 with some assignment still remaining in the $\vec{A}_i$ vertex. In this case the

second vertex in C_i would take only what remains of the assignment.

Algorithm 1 Form Flow Network Edges for Cluster C_i

$\vec{A}_i$ are the points $j \in \phi^{-1}(i)$ in non-increasing order of p_j
initialize array $\vec{a}$ of size $|C_i|$ to zeros, and set $s = 1$
put the assignment x_{ij} for each point j in $\vec{A}_i$ in $\vec{z}_i$ according the vertex order in $\vec{A}_i$
for $q = 1$ to $|C_i|$ **do**
  $\vec{a}(q) = \vec{a}(q) + x_{i\vec{A}_i(s)}$, and add edge $(\vec{A}_i(s), q)$
  $\vec{z}_i(s) = 0$
  $s = s + 1$ {Move to the next vertex}
  repeat
    $\text{valueToAdd} = min(1 - \vec{a}(q), \vec{z}_i(s))$
    $\vec{a}(q) = \vec{a}(q) + \text{valueToAdd}$, and add edge $(\vec{A}_i(s), q)$
    $\vec{z}_i(s) = \vec{z}_i(s) - \text{valueToAdd}$
    if $\vec{z}_i(s) = 0$ **then**
      $s = s + 1$
    end if
  until $\vec{a}(q) = 1$ or $s > |\vec{A}_i|$ {until we have accumulated 1 or ran out of vertices}
end for

We denote the set of edges that connect all points in $\mathcal{C}$ to points in C_i by $E_{\mathcal{C},C_i}$. Also, let $V_{\text{flow}} = \mathcal{C} \cup (\cup_{i \in S} C_i) \cup S \cup \{t\}$ and $E_{\text{flow}} = E_{\mathcal{C},C_i} \cup E_{C_i,S} \cup E_{S,t}$, where $E_{C_i,S}$ has an edge from every vertex in C_i to the corresponding center $i \in S$. Finally $E_{S,t}$ has an edge from every vertex i in S to the sink t if $\sum_{j \in \mathcal{C}} x_{ij} > \left\lfloor \sum_{j \in \mathcal{C}} x_{ij} \right\rfloor$. The demands, capacities, and costs of the network are:

- **Demands**: Each $v \in \mathcal{C}$ has demand $d_v = -1$ (a supply of 1), $d_u = 0$ for each $u \in C_i$, $d_i = \left\lfloor \sum_{j \in \mathcal{C}} x_{ij} \right\rfloor$ for each $i \in S$. Finally t has demand $d_t = |\mathcal{C}| - \sum_{i \in S} d_i$.
- **Capacities**: All edge capacities are set to 1.
- **Costs**: All edges have cost 0, expect the edges in $E_{\mathcal{C},C_i}$ where $\forall (v,u) \in E_{\mathcal{C},C_i}, d(v,u) = d(v,i)$ for the k-median and $d(v,u) = d^2(v,i)$. For the k-center, either setting suffices.

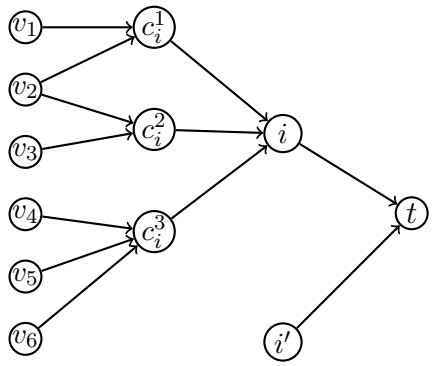


Figure 2.1: Network flow construction.

See Figure 2.1 for an example. It is clear that the entire demand is $|\mathcal{C}|$ and that this is the maximum possible flow. The LP solution attains that flow. Further, since the demands, capacities and distances are integers, an optimal integral minimum-cost flow can be found in polynomial time. If $\bar{x}_{ij}$ is the integer assignment that resulted from the flow computation, then violations are as follows:

Theorem 2.2.3. *(1) The number of vertices assigned to a cluster (cluster size) is violated by at most 1, i.e.* $|\sum_{j\in\mathcal{C}} \bar{x}_{ij} - \sum_{j\in\mathcal{C}} x_{ij}| \leq 1$. *(2) The fairness violation is at most* 2*, i.e.* $|\sum_{j\in\mathcal{C}} \bar{x}_{ij}p_j - \sum_{j\in\mathcal{C}} x_{ij}p_j| \leq 2$.

Proof. Part (1) follows by the demands and capacities set by the network flow scheme. To prove part (2), then note that for a given center i, every vertex $q \in C_i$ is assigned some vertices and adds value $\sum_{j\in\phi^{-1}(i,q)} p_j x_{ij}^q$ to the entire average (expected) value of cluster i where $\phi^{-1}(i,q)$ refers to the subset in $\phi^{-1}(i)$ assigned to q. After the rounding, $\sum_{j\in\phi^{-1}(i,q)} p_j x_{ij}^q$ will become $\sum_{j\in\phi^{-1}(i,q)} p_j \bar{x}_{ij}^q$. Denoting $\max_{j\in\phi^{-1}(i,q)} p_j$ and $\min_{j\in\phi^{-1}(i,q)} p_j$ by $p_{q,i}^{max}$ and $p_{q,i}^{min}$, respectively.

The following bounds the maximum violation:

$$
\begin{aligned}
&\sum_{q=1}^{|C_i|}\Big(\sum_{j\in\phi^{-1}(i,q)} p_j\bar{x}_{ij}^q\Big) - \sum_{q=1}^{|C_i|}\Big(\sum_{j\in\phi^{-1}(i,q)} p_j x_{ij}^q\Big)\\
&= \sum_{q=1}^{|C_i|}\sum_{j\in\phi^{-1}(i,q)}\Big(p_j\bar{x}_{ij}^q - p_j x_{ij}^q\Big)\\
&\leq p_{|C_i|,i}^{max} + \sum_{q=1}^{|C_i|-1} p_{q,i}^{max} - p_{q,i}^{min}\\
&= p_{|C_i|,i}^{max} + \Big(p_{1,i}^{max} - p_{1,i}^{min}\Big) + \Big(p_{2,i}^{max} - p_{2,i}^{min}\Big)\\
&+ \Big(p_{3,i}^{max} - p_{3,i}^{min}\Big) + \cdots + \Big(p_{|C_i|-1,i}^{max} - p_{|C_i|-1,i}^{min}\Big)\\
&\leq p_{|C_i|,i}^{max} + \Big(p_{1,i}^{max} - p_{1,i}^{min}\Big) + \Big(p_{1,i}^{min} - p_{2,i}^{min}\Big)\\
&+ \Big(p_{2,i}^{min} - p_{3,i}^{min}\Big) + \cdots + \Big(p_{|C_i|-2,i}^{min} - p_{|C_i|-1,i}^{min}\Big)\\
&\leq p_{|C_i|,i}^{max} + p_{1,i}^{max} - p_{|C_i|-1,i}^{min}\\
&\leq 2 - 0 = 2
\end{aligned}
$$

where we invoked the fact that $p_{k,i}^{max} \leq p_{k-1,i}^{min}$. By a similar argument it can be shown that the maximum drop is -2. □

Our rounding scheme results in a violation for the two color probabilistic case that is at most 2. We show a lower bound of at least $\frac{1}{2}$ for any rounding scheme applied to the resulting solution.

Theorem 2.2.4. *Any rounding scheme applied to the resulting solution has a fairness constraint violation of at least $\frac{1}{2}$ in the worst case.*

Proof. Consider the following instance (in Figure 2.2) with 5 points. Points 2 and 4 are chosen

as the centers and both clusters have the same radius. The entire set has an expected value of: $\frac{2(0)+2(\frac{3}{4})+1}{2+2+1} = \frac{\frac{5}{2}}{5} = \frac{1}{2}$. If the upper and lower values are set to $u = l = \frac{1}{2}$, then the fractional assignments for cluster 1 can be: $x_{21} = 1, x_{22} = 1, x_{23} = \frac{1}{2}$, leading to average color $\frac{\frac{3}{4}+0+\frac{1}{2}}{1+1+\frac{1}{2}} = \frac{1}{2}$. For cluster 2 we would have: $x_{43} = \frac{1}{2}, x_{44} = 1, x_{45} = 1$ and the average color is $\frac{(\frac{3}{4}+\frac{1}{2})}{\frac{5}{2}} = \frac{\frac{5}{4}}{\frac{5}{2}} = \frac{1}{2}$. Only assignments x_{23} and x_{43} are fractional and hence will be rounded. WLOG assume that $x_{23} = 1$ and $x_{43} = 0$. It follows that the change (violation) in the assignment $\sum_j p_j x_{ij}$ for a cluster i will be $\frac{1}{2}$. Consider cluster 1, the resulting color is $\frac{3}{4} + 1 = \frac{7}{4}$, the change is $|\frac{7}{4} - \frac{5}{4}| = \frac{1}{2}$. Similarly, for cluster 2 the change is $|\frac{5}{4} - \frac{3}{4}| = \frac{1}{2}$.

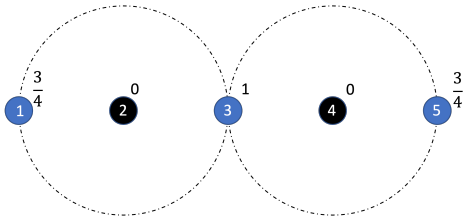


Figure 2.2: Points 2 and 4 have been selected as centers by the integer solution. Each points has its probability value written next to.

□

2.2.1.2 Algorithms for the Multiple Color Case Under a Large Cluster Assumption:

First, we point out that for the multi-color case, the algorithm is based on the assumption that the cluster size is large enough. Specifically:

Assumption 2.2.5. *Each cluster in the optimal solution should have size at least* $L = \Omega(n^r)$ *where* $r \in (0, 1)$.

We firmly believe that the above is justified in real datasets. Nonetheless, the ability to

manipulate the parameter r, gives us enough flexibility to capture all occurring real-life scenarios.

Theorem 2.2.6. *If Assumption 2.2.5 holds, then independent sampling results in the amount of color for each clusters to be concentrated around its expected value with high probability.*

Proof. The proof follows by invoking a Chernoff bound (see [14] for the full proof). □

Given Theorem 2.2.6 our solution essentially forms a reduction from the problem of probabilistic fair clustering $\mathrm{PFC}(k,p)$ to the problem of deterministic fair clustering with lower bounded cluster sizes which we denote by $\mathrm{DFC}_{\mathrm{LB}}(k,p,L)$ (the color assignments are known deterministically and each cluster is constrained to have size at least L). Our algorithm (2) in-

Algorithm 2 Algorithm for Large Cluster $\mathrm{PFC}(k,p)$

Input: $\mathcal{C}, d, k, p, L, \{(l_h, u_h)\}_{h\in\mathcal{H}}$
Relax the upper and lower by ϵ: $\forall\, h \in \mathcal{H}$, $l_h \leftarrow l_h(1-\epsilon)$ and $u_h \leftarrow u_h(1+\epsilon)$
For each point $j \in \mathcal{C}$ sample its color independently according to p_j^h
Solve the deterministic fair clustering problem with lower bounded clusters $\mathrm{DFC}_{\mathrm{LB}}(k,p,L)$ over the generated instance and return the solution.

volves three steps. In the first step, the upper and lower bounds are relaxed since -although we have high concentration guarantees around the expectation- in the worst case the expected value may not be realizable (could not be an integer). Moreover the upper and lower bounds could coincide with the expected value causing violations of the bounds with high probability.

After the color assignments are sampled independently, we have to solve the deterministic fair clustering with lower bounded cluster sizes problem $\mathrm{DFC}_{\mathrm{LB}}$. We can establish the following theorem for $\mathrm{DFC}_{\mathrm{LB}}$:

Theorem 2.2.7. *Given an α approximation algorithm for the color blind clustering problem $\mathrm{Cluster}(k,p)$ and a γ violating algorithm for the fair assignment problem with lower bounded*

cluster sizes FA-PFC-LB(S, p, L)*, a solution with approximation ratio* $\alpha + 2$ *and violation at most* γ *can be achieved for the deterministic fair clustering problem with lower bounded cluster size* $\mathrm{DFC}_{\mathrm{LB}}(k, p)$.

Proof. The proof follows by arguments similar to the proof of Theorem 2.2.1. □

Having established the above theorem for $\mathrm{DFC}_{\mathrm{LB}}(k, p)$ we only have to solve the fair assignment problem with lower bounded cluster sizes FA-PFC-LB(S, p, L). The approach is similar to the two color case. Specifically, we have to solve the following LP:

$$\min_{S:|S|\leq k,\phi} L_p^k(\mathcal{C}) \tag{2.6a}$$

$$\text{s.t. } \forall i \in S : (1-\epsilon) l_h |\,\mathcal{C}_i\,| \leq |\,\mathcal{C}_{i,h}\,| \leq (1+\epsilon) u_h |\,\mathcal{C}_i\,| \tag{2.6b}$$

$$\forall i \in S : |\,\mathcal{C}_i\,| \geq L \tag{2.6c}$$

Note that the bounds are relaxed by ϵ and a lower bound L is required on the cluster size. Since a center might be assigned no points and the above LP imposes a lower bound on each center, we run multiple versions of the LP with each closing a set of centers (removing them from S). This leads to a fixed-parameter tractable solution where the run-time is $O(2^k \mathrm{poly}(n))$. Once the LP is solved a min-cost rounding scheme from [1] is used. Now we establish the final approximation ratio of $\alpha + 2$:

Theorem 2.2.8. *Given an instance of the probabilistic fair clustering problem* $\mathrm{PFC}(k, p)$ *where assumption 2.2.5 holds, then with high probability algorithm 2 results in a solution with violation at most* ϵ *and approximation ratio* $(\alpha + 2)$ *in* $O(2^k poly(n))$ *time.*

Proof. First, given an instance $\mathcal{I}_{\mathrm{PFC}}$ of probabilistic fair clustering with optimal value $\mathrm{OPT}_{\mathrm{PFC}}$ the clusters in the optimal solution would with high probability be a valid solution for the deterministic setting, as showed in Theorem 2.2.6. Therefore, the resulting deterministic instance would have $\mathrm{OPT}_{\mathrm{DFC}_{\mathrm{LB}}} \leq \mathrm{OPT}_{\mathrm{PFC}}$. Hence, the algorithm will return a solution with cost at most $(\alpha+2)\,\mathrm{OPT}_{\mathrm{DFC}_{\mathrm{LB}}} \leq (\alpha+2)\,\mathrm{OPT}_{\mathrm{PFC}}$.

For the solution $\mathrm{SOL}_{\mathrm{DFC}_{\mathrm{LB}}}$ returned by the algorithm, each cluster is of size at least L, and the Chernoff bound guarantees that the violation in expectation is at most ϵ with high probability.

The run-time comes from the fact that $\mathrm{DFC}_{\mathrm{LB}}$ is solved in $O(2^k \mathrm{poly}(n))$ time. □

2.2.2 Experiments

We now evaluate the performance of our algorithms over a collection of real-world datasets. We give experiments in the two color case (§2.2.2.1) as well as under the large cluster assumption (§2.2.2.2). We include experiments for the k-means case here.

Color-Blind Clustering. The color-blind clustering algorithms we use are as follows.

- [41] gives a 2-approximation for k-center.
- We use `Scikit-learn`'s k-means++ module.
- We use the 5-approximation algorithm due to [43] modified with D-sampling [44] according to [40].

Generic-Experimental Setup and Measurements. For a chosen dataset, a given color h would have a proportion $f_h = \frac{|v \in \mathcal{C}\,|\chi(v)=h|}{|\,\mathcal{C}\,|}$. Following [40], the lower bound is set to $l_h = (1-\delta)r_h$ and the upper bound is to $u_h = \frac{f_h}{(1-\delta)}$. For metric membership, we similarly have $f = \frac{\sum_{j \in \mathcal{C}} r_j}{|\,\mathcal{C}\,|}$ as

the proportion, $l = (1-\delta)f$ and $u = \frac{f}{1-\delta}$ as the lower and upper bound, respectively. We set $\delta = 0.2$, as [40] did, unless stated otherwise.

For each experiment, we measure the price of fairness $\text{PoF} = \frac{\text{Fair Solution Cost}}{\text{Color-Blind Cost}}$. We also measure the maximum additive violation γ as it appears in inequality 2.4.

2.2.2.1 Two Color Case

Here we test our algorithm for the case of two colors with probabilistic assignment. We use the **Bank** dataset [45] which has 4,521 data points. We choose marital status, a categorical variable, as our fairness (color) attribute. To fit the binary color case, we merge single and divorced into one category. Similar to the supervised learning work due to [46], we make **Bank**'s deterministic color assignments probabilistic by independently perturbing them for each point with probability p_{noise}. Specifically, if j originally had color c_j, then now it has color c_j with probability $1 - p_{\text{noise}}$ instead. To make the results more interpretable, we define $p_{\text{acc}} = 1 - p_{\text{noise}}$. Clearly, $p_{\text{acc}} = 1$ corresponds to the deterministic case, and $p_{\text{acc}} = \frac{1}{2}$ corresponds to completely random assignments.

First, in Fig. 2.3(a), we see that the violations of the color-blind solution can be as large as 25 whereas our algorithm is within the theoretical guarantee that is less than 1. In Fig. 2.3(b), we see that in spite of the large violation, fairness can be achieved at a low relative efficiency loss, not exceeding 2% ($\text{PoF} \leq 1.02$).

How does labeling accuracy level p_{acc} impact this problem? Fig. 2.4 shows p_{acc} vs PoF for $\delta = 0.2$ and $\delta = 0.1$. At $p_{\text{acc}} = \frac{1}{2}$, color assignments are completely random and the cost is, as expected, identical to color-blind cost. As p_{acc} increases, the colors of the vertices become more

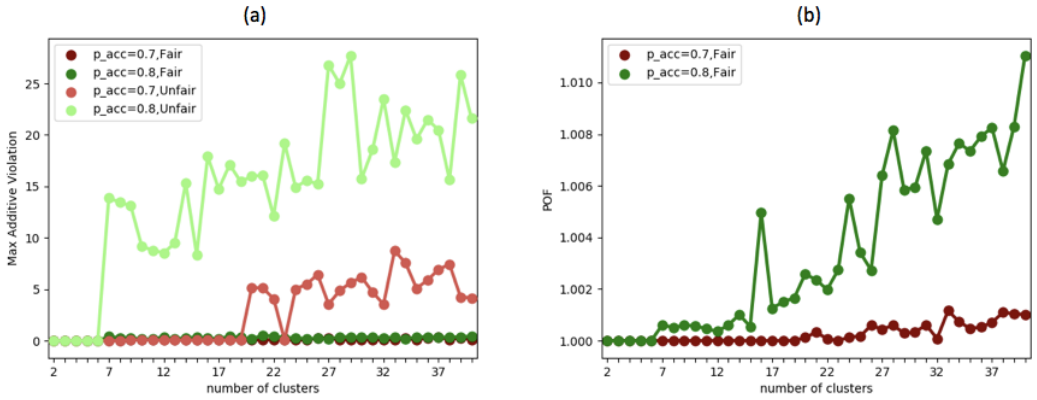


Figure 2.3: For $p_{\text{acc}} = 0.7$ & $p_{\text{acc}} = 0.8$, showing (a): #clusters vs. maximum additive violation; (b): #clusters vs. PoF .

differentiated, causing PoF to increase, eventually reaching the maximum at $p_{\text{acc}} = 1$ which is the deterministic case.

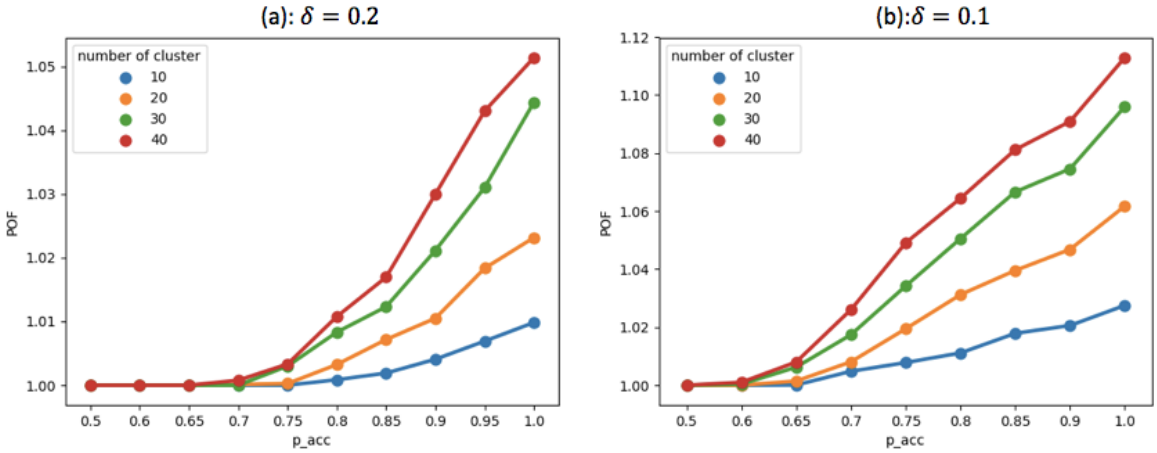


Figure 2.4: Plot showing p_{acc} vs PoF, (a):$\delta = 0.2$ and (b):$\delta = 0.1$.

Next, we test against an "obvious" strategy when faced with probabilistic color labels: simply *threshold* the probability values, and then run a deterministic fair clustering algorithm. Fig. 2.5(a) shows that this may indeed work for guaranteeing fairness, as the proportions may be satisfied with small violations; however, it comes at the expense of a much higher PoF. Fig. 2.5(b) supports this latter statement: our algorithm can achieve the same violations with smaller PoF.

Further, running a deterministic algorithm over the thresholded instance may result in an infeasible problem.[1]

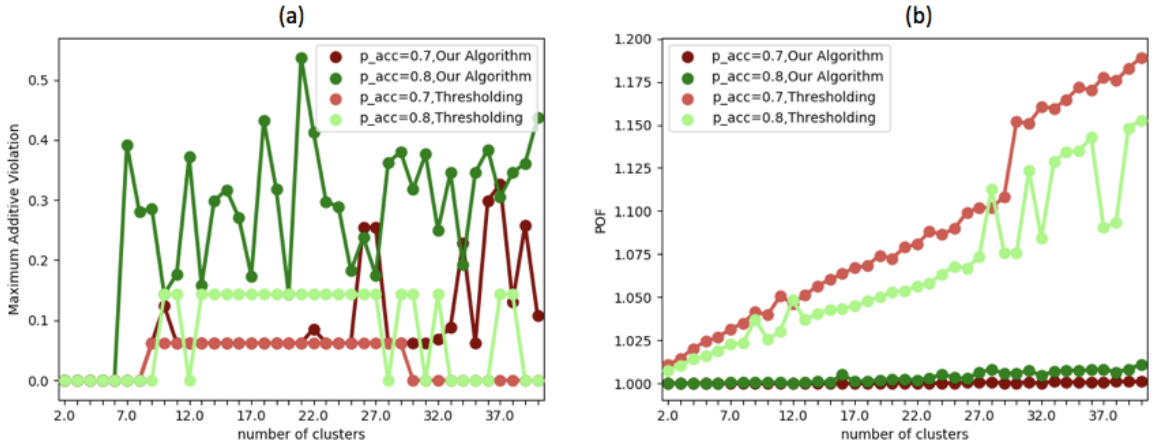


Figure 2.5: Comparing our algorithm to thresholding followed by deterministic fair clustering: (a)maximum violation, (b) PoF.

2.2.2.2 The Large Cluster Assumption

Here we test our algorithm for the case of probabilistically assigned multiple colors under Assumption 2.2.5, which addresses cases where the optimal clustering does not include pathologically small clusters. We use the **Census1990** [47] dataset. We note that **Census1990** is large, with over 2.4 million points. We use age groups (attribute `dAge` in the dataset) as our fairness attribute, which yields 7 age groups (colors).[2] We then sample 100,000 data points and use them to train an SVM classifier[3] to predict the age group memberships. The classifier achieves an accuracy of around 68%. We use the classifier to predict the memberships of another 100,000 points

[1] An intuitive example of infeasibility: consider the two color case where $p_j = \frac{1}{2} + \epsilon, \forall\, j \in \mathcal{C}$ for some small positive ϵ. Thresholding drastically changes the overall probability to 1; therefore no subset of points would have proportion around $\frac{1}{2} + \epsilon$.

[2] Group 0 is extremely rare, to the point that it violates the "large cluster" assumption for most experiments; therefore, we merged it with Group 1, its nearest age group.

[3] We followed standard procedures and ended up with a standard RBF-based SVM; the accuracy of this SVM is somewhat orthogonal to the message of this paper, and rather serves to illustrate a real-world, noisy labeler.

not included in the training set, and sample from that to form the probabilistic assignment of colors. Although as stated earlier we should try all possible combinations in closing and opening the color-blind centers, we keep all centers as they are. It is expected that this heuristic would not lead to a much higher cost if the dataset and the choice of the color-blind centers is sufficiently well-behaved.

Fig. 2.6 shows the output of our large cluster algorithm over 100,000 points and $k = 5$ clusters with varying lower bound assumptions. Since the clusters here are large, we normalize the additive violations by the cluster size. We see that our algorithm results in normalized violation that decrease as the lower bound on the cluster size increases. The PoF is high relative to our previous experiments, but still less than 50%.

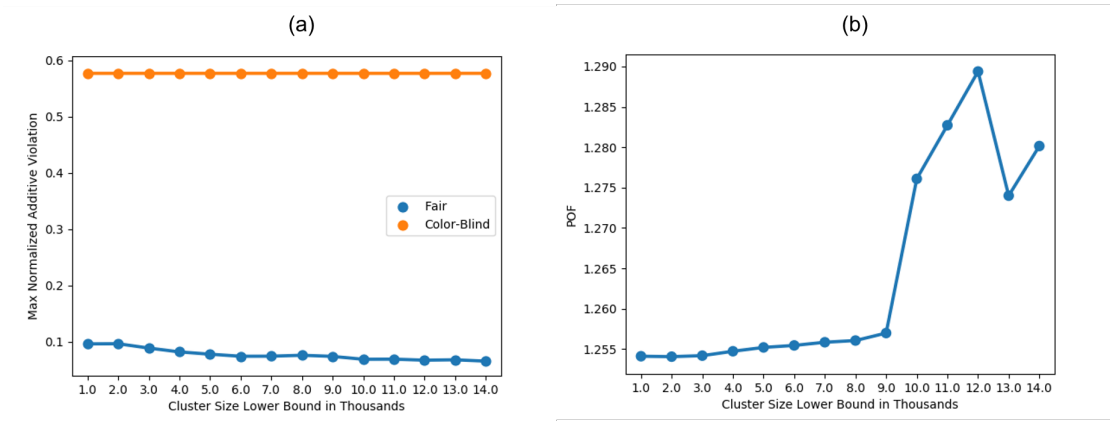


Figure 2.6: Plot showing the performance of our independent sampling algorithm over the **Census1990** dataset for $k = 5$ clusters with varying values on the cluster size lower bound:(a)maximum violation normalized by the cluster size, (b)the price of fairness.

2.3 Fair Clustering Under a Bounded Cost

An acknowledged fact in fair clustering—and, indeed, in many allocation and matching settings—is that the fairness (e.g., proportion) constraint could cause degradation in the clustering

objective [48, 49]. A point may be assigned to a further away center (cluster) to satisfy the proportion constraint [8]. The degradation in the objective due to the imposed fairness constraint is called the *price of fairness* (PoF), mathematically defined as $\text{PoF} = \frac{\textit{cost of fair solution}}{\textit{cost of agnostic solution}}$.

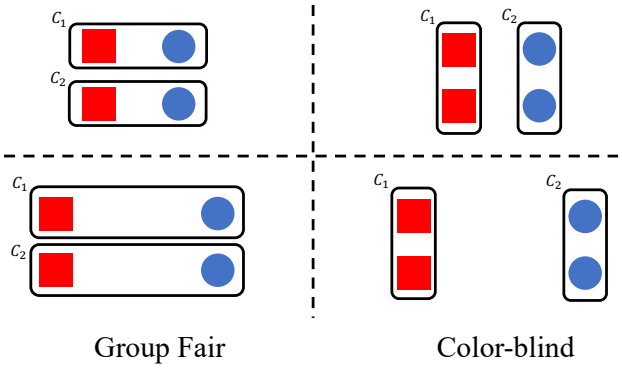


Figure 2.7: Comparison between group fair (left) and color-blind (right) clustering. Unlike color-blind clusters, group fair clusters may combine faraway points (bottom-left).

Unlike some examples in the literature [48, 50], the price of fairness in the case of fair clustering is unbounded, as seen in Figure 2.7. By enforcing a form of group fairness requiring an even split across colors in each cluster, a fair clustering algorithm would perform arbitrarily poorly as the two groups of points separate in space, while a "color-blind" algorithm would remain unchanged (bottom-left and bottom-right of Figure 2.7, respectively). The possibly unbounded increase in the clustering cost (unbounded price of fairness) indicates that fair clustering can yield clusters consisting of points that are far apart in the metric space instead of combining nearby points—often the main motivation behind clustering in machine learning and data analysis. Furthermore, the legal notion of disparate impact does not force an organization to output a fair clustering if it can justify an unfair one due to "business necessity," i.e., potential loss in quality [51,52]. This possible conflict between the clustering objective and the fairness constraint indicates the need for fair clustering algorithms that operate in a setting where the clustering cost cannot exceed a pre-set upper bound.

The fundamental idea of fair clustering under a bounded cost (**FCBC**) is to minimize a measure of unfairness subject to an upper bound on the clustering cost:

$$\min \text{ Unfairness} \tag{2.7a}$$

$$\text{s.t. Clustering Cost} \leq \text{Given upper bound} \tag{2.7b}$$

Next, we transform (2.7a) and (2.7b) above into a clear mathematical optimization problem.

The Constraint (2.7b): The clustering cost is $\left(\sum_{j\in\mathcal{C}} d^p(j,\phi(j))\right)^{1/p}$. Let U denote the exogenous upper bound on clustering cost. Then, (2.7b) becomes $\left(\sum_{j\in\mathcal{C}} d^p(j,\phi(j))\right)^{1/p} \leq U$. Note that for the case of the k-center where $p=\infty$, the constraint reduces to a simpler form, specifically $\forall j \in \mathcal{C}, d(j,\phi(j)) \leq U$.

The Objective (2.7a): In prior work, a given clustering is considered fair if for each cluster, the proportions of each color lie within pre-specified lower and upper bounds, i.e.: $\forall i \in S, \forall\, h \in \mathcal{H} : \beta_h |\,\mathcal{C}_i\,| \leq |\,\mathcal{C}_i^h\,| \leq \alpha_h |\,\mathcal{C}_i\,|$. However, bounding the clustering cost may make it impossible to have a fair feasible solution. Therefore, we instead set a measure of unfairness for each color and try to minimize this measure. Let Δ_h denote the worst proportional violation across the clusters for a color h. Specifically, for a given clustering, $\Delta_h \in [0,1]$ is the minimum non-negative value such that:

$$\forall i \in S : (\beta_h - \Delta_h)|\,\mathcal{C}_i\,| \leq |\,\mathcal{C}_i^h\,| \leq (\alpha_h + \Delta_h)|\,\mathcal{C}_i\,|. \tag{2.8}$$

Clearly, if $\Delta_h = 0$, then color h is within the desired proportion in every cluster. Having set Δ_h to be a measure of the unfair treatment that group h receives, we are faced with the question of setting the fairness objective, for which there are many reasonable options. We consider two

prominent and intuitive fairness objectives [53]:

$$\text{GROUP-UTILITARIAN} = \min \sum_{h \in \mathcal{H}} \Delta_h \quad , \quad \text{GROUP-EGALITARIAN} = \min \max_{h \in \mathcal{H}} \Delta_h$$

The GROUP-UTILITARIAN objective minimizes the sum of proportional violations for all of the colors, treating all points of a specific color as a single player in a game. The GROUP-EGALITARIAN objective minimizes the maximum proportional violation across the colors. We also consider a more generalized version of the GROUP-EGALITARIAN objective, the GROUP-LEXIMIN objective. Like GROUP-EGALITARIAN, the GROUP-LEXIMIN objective minimizes the maximum (worst) violation, but it goes further to minimizes the second-worst violation, then the third-worst violation, and so on until no further improvement can be made. We now state the fair clustering under a bounded cost problem ($\mathbf{FCBC}$):

$$\min_{S:|S| \leq k, \phi} \text{UNFAIRNESS-OBJECTIVE} \tag{2.9a}$$

$$\text{s.t.} \qquad \Big(\sum_{j \in \mathcal{C}} d^p(j, \phi(j)) \Big)^{1/p} \leq U \tag{2.9b}$$

where the UNFAIRNESS-OBJECTIVE could equal GROUP-UTILITARIAN, GROUP-EGALITARIAN, or GROUP-LEXIMIN. Similar to the fair assignment $\mathbf{FA}$ problem, we may define the fair assignment under a bounded cost ($\mathbf{FABC}$) problem as:

$$\min_{\phi} \text{UNFAIRNESS-OBJECTIVE} \tag{2.10a}$$

$$\text{s.t.} \qquad \Big(\sum_{j \in \mathcal{C}} d^p(j, \phi(j)) \Big)^{1/p} \leq U \tag{2.10b}$$

where similarly the optimization is over the assignment function ϕ while the set of centers S is fixed.

2.3.1 Hardness of FCBC & FABC

We now establish the hardness of fair clustering under a bounded cost **FCBC** and fair assignment under a bounded cost **FABC**. We note that these hardness results follow for all objectives (GROUP-UTILITARIAN, GROUP-EGALITARIAN, and GROUP-LEXIMIN).

Theorem 2.3.1. *Fair clustering under a bounded cost* **FCBC** *and fair assignment under a bounded cost* **FABC** *are NP-hard.*

Proof. We first start with the following lemma about fair clustering and fair assignment. Note that fair clustering and fair assignment problems are $\mathrm{PFC}(k, p)$ and $\text{FA-PFC}(S, p)$ (from the previous subsection), respectively. Although here we are concerned only with the special case where the color values are known deterministically:

Lemma 2.3.2. *The fair clustering and fair assignment problems are NP-hard.*

Proof. Since fair clustering problems, i.e. fair k-(center, median, or means) generalize their NP-hard classical counterparts, i.e. the k-(center, median, or means) clustering, it follows that fair clustering problems are also NP-hard.

The hardness of the fair assignment problem was established by [1] for k-center clustering. Here we show that fair assignment is NP-hard for k-median and k-means clustering as well.

First, following Section 4 of [1], the reduction is from the Exact Cover by 3-Sets (X3C). In Exact Cover by 3-Sets, we have a universal set of elements $\mathcal{U}$ with $|\mathcal{U}| = 3r$ where r is a positive integer and a set $\mathcal{F}$ whose elements are subsets of $\mathcal{U}$. The problem is to decide if there exists a set $\mathcal{F}'$ such that $\mathcal{F}' \subseteq \mathcal{F}$ and each element in $\mathcal{U}$ is included exactly once in one set in $\mathcal{F}'$.

The reduction is done by creating the following graph (see Figure 2.8 for an example). In

the lowest level we have the elements e of the set $\mathcal{U}$ each represented with a blue vertex. In the higher level we have the sets in $\mathcal{F}$ each represented with a blue vertex. We draw edges between vertices in $e \in \mathcal{U}$ and vertices in $F \in \mathcal{F}$ if and only if the element $e \in F$. For set F in $\mathcal{F}$ we add 3 auxiliary blue vertices which are connected to it through an edge. Finally, we add a set $\mathcal{T}$ of red vertices where $|\mathcal{T}| = \frac{|\mathcal{U}|}{3} = r$ in the highest level where each of those vertices is connected through an edge to every vertex in the set $\mathcal{F}$.

The distance function puts a cost of zero if the distance is between identical vertices and a cost of one between vertices connected through an edge. For vertices with no edges between them, the distance is the minimum distance found according to this graph by calculating the minimum cost path. This means that the distance between the blue auxiliary vertices and a center which is not their parent center is 3 (the path from the vertex to the associated center to an element in $\mathcal{T}$, then the specified center).

In fair assignment, the set of centers is already chosen. We choose the set of centers to be the elements of $\mathcal{F}$. Therefore, the number of centers $k = r$. Further, it is clear that this is a two color problem, we set the lower and upper bounds for the red color to $\beta_{\text{red}} = \alpha_{\text{red}} = \frac{1}{4}$. It follows that $\beta_{\text{blue}} = \alpha_{\text{blue}} = \frac{3}{4}$, i.e. the ratio of red to blue vertices is $1 : 3$.

We note the following claim:

Claim 2.3.3. *Given the constructed graph with the set of centers being $\mathcal{F}$, the minimum clustering cost is lower bounded by 1 for the k-center problem and $n - k$ for the k-median and k-means.*

Proof. First we note the following fact:

Fact 1. *$\forall u, v \in G$ where u and v are distinct, we have that $d(u, u) = d(v, v) = 0$ and $d(u, v) \geq 1$ if $u \neq v$.*

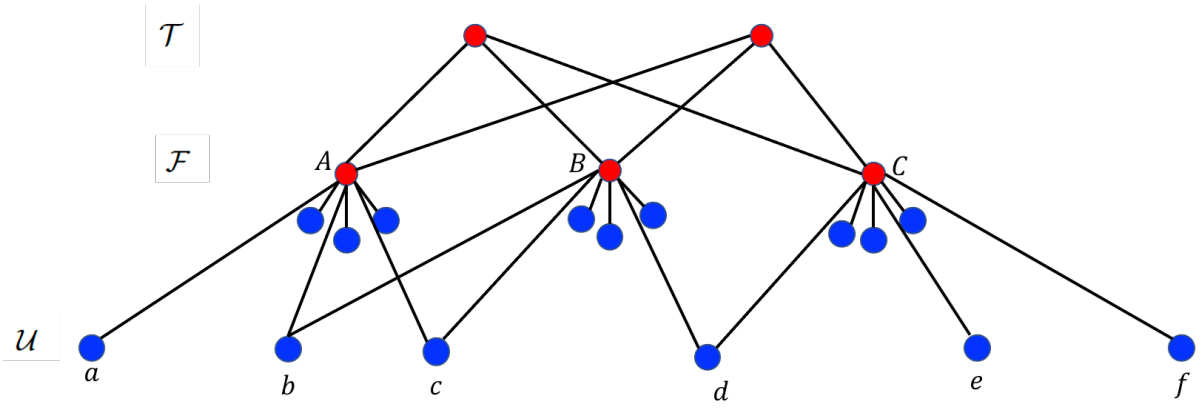


Figure 2.8: Figure follows the example of [1]. We show the fair assignment resulting graph, from the given Exact Cover by 3-Sets example where we have $\mathcal{U} = \{a, b, c, d, e, f\}$ and $\mathcal{F} = \{A = \{a, b, c\}, B = \{b, c, d\}, C = \{d, e, f\}\}$.

k-center: Since the number of points is greater than the number of centers it follows that there exists a point u which will be assigned to another vertex v and therefore $d(u, v) \geq 1$.

k-median and k-means: Denoting the assignment function (assigning vertices to centers) by ϕ, the set of centers by S, and the integer p where $p = 1$ for the k-median and $p = 2$ for the k-means, we have that:

$$\begin{aligned}
\sum_{v \in G} d^p(v, \phi(v)) &= \sum_{v \in S} d^p(v, \phi(v)) + \sum_{v \in G-S} d^p(v, \phi(v)) \\
&\geq 0 + \sum_{v \in G-S} d^p(v, \phi(v)) \\
&\geq 0 + \sum_{v \in G-S} 1^p \\
&\geq \sum_{v \in G-S} 1 \\
&= n - k
\end{aligned}$$

where the above follows from Fact 1. □

Therefore, we have:

Claim 2.3.4. *If there exists an exact cover, then the fair assignment problem can have a* $1:3$ *red to blue vertex ratio and at a cost of 1 for the* k*-center and a cost of* $n-k$ *for the* k*-median and* k*-means.*

Proof. We translate the exact cover by 3-sets solution to the constructed graph. Each chosen set in exact cover $\mathcal{F}'$ will have the 3 corresponding elements from $\mathcal{U}$ assigned to its center, along with its 3 auxiliary vertices and 1 vertex from $\mathcal{T}$. If the set was not chosen in the exact cover, then it will have only its 3 auxiliary vertices assigned to it.

This clearly matches the lower bound on the cost function from claim (2.3.3) for each clustering objective. Further, it is also clear that the $1:3$ red to blue color ratio is preserved in each cluster. □

Claim 2.3.5. *If there exists a fair assignment solution with* $1:3$ *red to blue proportion and whose cost is* 1 *for the* k*-center and* $(n-k)$ *for the* k*-median and* k*-means, there exists a solution to the exact cover by 3-sets problem.*

Proof. The costs of 1 and $(n-k)$ for the k-center and k-median/mean respectively can only be achieved by assigning elements $e \in \mathcal{U}$ to a center that they have an edge between. Similarly, all of the blue auxiliary vertices have to be assigned to their parent. Further to achieve the $1:3$ red to blue ratio, a center will either choose 3 elements from $\mathcal{U}$ and therefore has to choose an element from $\mathcal{T}$ to satisfy the proportion. Or a center will not choose any element from $\mathcal{U}$ and in that case it would not need to pick an element from $\mathcal{T}$ to satisfy the proportion. □

□

With the above lemma we can now easily prove the theorem. First, the hardness of fair clustering under a bounded cost $\mathbf{FCBC}$ simply follows by setting the upper bound to $U = \mathrm{OPT}_{\mathbf{FC}}$ where $\mathrm{OPT}_{\mathbf{FC}}$ is the optimal value of fair clustering $\mathbf{FC}$. An optimal solution to fair clustering would achieve the optimal value of 0 for all possible fairness objectives of $\mathbf{FCBC}$ and would have a cost $\mathrm{OPT}_{\mathbf{FC}} \leq U$.

Conversely, an optimal solution for $\mathbf{FCBC}$ would have a proportional violation of zero for all colors (therefore it is fair). Moreover, its cost would not exceed $U = \mathrm{OPT}_{\mathbf{FC}}$. Therefore, it is an optimal solution for fair clustering.

By the above, a solution is optimal for a fair clustering if and only if it is an optimal solution to the corresponding $\mathbf{FCBC}$ instance with $U = \mathrm{OPT}_{\mathbf{FC}}$. It follows that since fair clustering is NP-hard, that fair clustering under a bounded cost $\mathbf{FCBC}$ is also NP-hard.

In a similar manner, by setting $U = \mathrm{OPT}_{\mathbf{FA}}$ the hardness of fair assignment under a bounded cost $\mathbf{FABC}$ can be established from the hardness of fair assignment. □

For a given clustering cost U, there are many clusterings (solutions) of cost not exceeding U. Let $\mathcal{S}_U$ be the set of those solutions, i.e. if $(S_t, \phi_t) \in \mathcal{S}_U$, then (S_t, ϕ_t) is a clustering with a cost that does not exceed U. Further, let L_t be the size of the smallest non-empty cluster[4] in the clustering (S_t, ϕ_t), then we define $L(U)$ to be the size of the smallest cluster across all clusterings of cost not exceeding U, i.e. $L(U) = \min_{(S_t,\phi_t)\in\mathcal{S}_U} L_t$. Clearly, for U_1 and U_2 such that $U_2 \geq U_1$, then $L(U_2) \leq L(U_1)$ since $\mathcal{S}_{U_1} \subseteq \mathcal{S}_{U_2}$. We can conclude the following fact from the definition of $L(U)$:

[4]An empty cluster is a cluster with no points assigned to it. This could happen if for example the assignment function ϕ does not map any point to a a given center including the center itself.

Fact 2. *For a given upper bound U, no clustering with cost less than or equal to U can have less than $L(U)$ many points in a non-empty cluster.*

We show that the quantity $L(U)$ plays a fundamental role. In fact, lower bounds on the additive approximation[5] for the proportional violations and fairness objectives are related to $L(U)$ as shown in the following theorem:

Theorem 2.3.6. *For a given instance of the* **FCBC** *or* **FABC** *problem with an arbitrary upper bound U, unless $P = NP$ no polynomial time algorithm can produce a solution with a cost not exceeding U that satisfies any of the following conditions:* ***(a)*** *The proportional violation of any color $h \in \mathcal{H}$ is $\Delta_h < \frac{1}{8L(U)}$.* ***(b)*** *The additive approximation for the* GROUP-UTILITARIAN *objective is less than $\frac{|\mathcal{H}|}{8L(U)}$.* ***(c)*** *The additive approximation for the* GROUP-EGALITARIAN *objective is less than $\frac{1}{8L(U)}$.*

Proof. We note that our derivation uses the reduction from X3C shown in the proof of Lemma (2.3.2) and the resulting graph shown in figure (2.8). We start by deriving a collection of useful claims:

Claim 2.3.7. *If $U = 1$ for the k-center objective or $U = n - k$ for the k-median and k-means objectives, then $L(U) = 4$ for all objectives. Further the only set of centers that can lead to a cost not exceeding U is $S = \mathcal{F}$.*

Proof. First it is clear that if we choose the set $\mathcal{F}$ to be the centers, i.e. $S = \mathcal{F}$, then if we route each point to one of its closest centers in $\mathcal{F}$, then we can have for the k-center we would have a cost of 1 since every point in the graph is at most a distance 1 from a point in $\mathcal{F}$. Further, for the

[5] An algorithm for a minimization problem with additive approximation $\mu > 0$, returns a value for the objective that is at most OPT $+\mu$ where OPT is the optimal value.

k-median and k-means objectives, the points $\mathcal{F}$ would be routed to themselves and every other point would be routed to one of its closest centers in $\mathcal{F}$ which is at a distance of 1, this leads to a cost of $(0)k + (1)(n-k) = n-k$, therefore choosing the $\mathcal{F}$ as the set of centers we can indeed satisfy the upper bound U for all objectives.

Now, consider another set of centers S' such that $\exists i \in S'$ and $i \notin \mathcal{F}$, i.e. we have at least one center not from $\mathcal{F}$. Let f be the point in $\mathcal{F}$ not selected in S'. For the k-center objective with $U = 1$, it follows that the blue auxiliary points of f have to be made as centers since every other point is at least a distance of 2 away from them, but each auxiliary point of f is made a center, then it follows that $|S' - \mathcal{F}| \geq 3$, i.e. at least two more points of $\mathcal{F}$ have not be selected as centers. We can invoke the argument again on the new auxiliary points to conclude that $|S' - \mathcal{F}| \geq 9$. Invoking the argument again, we will see get that $|S' - \mathcal{F}| \geq 3k$ which is infeasible since $|S' - \mathcal{F}| \leq |S'| \leq k$. Therefore, for the k-center with $U = 1$, we must have $S = \mathcal{F}$. Now having proven that $S = \mathcal{F}$ and since $U = 1$, it follows that the smallest cluster size is 4 formed by mapping the center in $S = \mathcal{F}$ to itself along with its auxiliary points, i.e. $L(U) = 4$ for the k-center.

For the k-median and k-means objectives with $U = n - k$, similiar to the k-center it is clear that every point which has not been selected as a center must have a center at a distance of at most 1 away. If we exclude one point $f \in \mathcal{F}$ from the set of centers, then its auxiliary points will each have to become centers to satisfy the upper bound cost of $U = n - k$, but this would mean that there are at least 2 more points in $\mathcal{F}$ that have been excluded. Following an argument similar to that of the k-center, we will have that the set of required centers would be at least $3k$ which is a contradiction. Therefore, the only possible choice of centers is $S = \mathcal{F}$. It follows as well that the smallest cluster size if 4 formed by mapping the center in $S = \mathcal{F}$ to itself along with

its auxiliary points, i.e. $L(U) = 4$ for the k-median and k-means objectives. □

Further, we define Δ^i_{red} and Δ^i_{blue} as the red and blue violations in the i^{th} cluster, respectively. Then we have the following claim:

Claim 2.3.8. *For the two color case of the above reduction, $\Delta^i_{red} = \Delta^i_{blue}$ and $\Delta_{red} = \Delta_{blue}$.*

Proof. for cluster i, consider the red and blue violations $\Delta^i_{\text{red}}, \Delta^i_{\text{blue}}$ at that cluster, then we have:

$$\Delta^i_{\text{red}} = |p^i_{\text{red}} - \frac{1}{3}| = |(1 - p^i_{\text{blue}}) - (1 - \frac{2}{3})| = |\frac{2}{3} - p^i_{\text{blue}}| = \Delta^i_{\text{blue}}$$

It is clear then that $\Delta_{\text{red}} = \max_{i \in [k]} \Delta^i_{\text{red}} = \max_{i \in [k]} \Delta^i_{\text{blue}} = \Delta_{\text{blue}}$ □

The following lemma follows immediately from the above:

Claim 2.3.9. *For the two color case of the above reduction we have*

$$\text{GROUP-UTILITARIAN} = 2\text{GROUP-EGALITARIAN}$$

Proof. $\text{GROUP-UTILITARIAN} = \Delta_{\text{red}} + \Delta_{\text{blue}} = 2\Delta_{\text{red}} = 2\text{GROUP-EGALITARIAN}$. □

We also note the following claim:

Claim 2.3.10. *For a given cluster i with set of points C_i, if the set of red points in the cluster C^{red}_i satisfy $\Delta^i_{red} = |\frac{|C^{red}_i|}{|C_i|} - \frac{1}{4}| < \frac{1}{4|C_i|}$, then cluster i has no violation.*

Proof. Suppose that $|\frac{|C^{\text{red}}_i|}{|C_i|} - \frac{1}{4}| < \frac{1}{4|C_i|}$, then it follows that $||C^{\text{red}}_i| - \frac{1}{4}|C_i|| < \frac{1}{4}$. Since $|C_i|$ is an integer it follows that $\frac{1}{4}|C_i|$ is of the form $m, m + \frac{1}{4}, m + \frac{1}{2},$ or $m + \frac{3}{4}$ where m is an integer. Further since $|C^{\text{red}}_i|$ is also an integer, the fact that $||C^{\text{red}}_i| - \frac{1}{4}|C_i|| < \frac{1}{4}$ implies that $|C^{\text{red}}_i| = \frac{1}{4}|C_i|$

and we have no violation for the red color in cluster i. Further, from Lemma 2.3.8 the blue violation equals the red violation and therefore we have no violation in cluster i. $\square$

Now we are ready to prove the main claims for the **FCBC** problem.

For the first claim, assume by contradiction that a polynomial time algorithm gave a solution of violation less than $\frac{1}{8L}$ and cost $\leq U$. Now, if we consider clusters i of size $|C_i|$ such that $4 \leq |C_i| \leq 8$, then it clear that since $\Delta^i_{\text{red}} \leq \Delta_{\text{red}} \leq \frac{1}{8L(U)}$, $\Delta^i_{\text{red}} \leq \frac{1}{4|C_i|}$ because $|C_i| \leq 8 \leq 2L(U)$, therefore there is no violation in these clusters by Claim 2.3.10.

Now consider a cluster of size greater than 8, (note by Claim 2.3.7 that $S = \mathcal{F}$) because of the upper bound U such clusters could only add points for the top row set $\mathcal{T}$ to the cluster which are all red, it clear that the more red points are added the greater the violation, if one additional red point is added, then for the best color proportions the cluster has a total of: 6 blues and 3 reds, which lead to a violation of $|\frac{1}{3} - \frac{1}{4}| = \frac{1}{12} > \frac{1}{8L(U)} = \frac{1}{32}$, therefore it is impossible for the algorithm to form such clusters as that would contradict the assumption that the algorithm obtains a violation $< \frac{1}{8L(U)}$ for each color. Therefore such clusters are not possible. This means that there is no violation in any cluster and that the problem has been solved optimally which by the NP-hardness is impossible unless $P = NP$.

Now the two remaining claims follow easily. By definition we have that GROUP-EGALITARIAN $= \max_{h \in \mathcal{H}} \Delta_h$. If GROUP-EGALITARIAN $< \frac{1}{8L(U)}$, then it follows that $\Delta_h < \frac{1}{8L(U)}$ for every color $h \in \mathcal{H}$ which by the first claim cannot happen unless $P = NP$.

Further, by Claim 2.3.9 GROUP-UTILITARIAN $= 2$ GROUP-EGALITARIAN, therefore if GROUP-UTILITARIAN $< \frac{|\mathcal{H}|}{8L(U)}$, then GROUP-EGALITARIAN $< \frac{1}{8L}$. which is impossible unless $P = NP$.

The same claims for the **FABC** problem can be proven by simply setting the set of centers $S = \mathcal{F}$ and the upper bound $U = 1$ for k-center and $n-k$ for the k-median/means, then following similar arguments. □

2.3.2 Algorithms for FCBC

Our main result for the **FCBC** problem is the following theorem which follows as a direct consequence of the guarantees of Theorems 2.3.12, 2.3.17, 2.3.15, 2.3.14, and 2.3.18:

Theorem 2.3.11. *For any clustering objective, given a bound U on the clustering cost, Algorithm 3 solves the fair clustering under a bounded cost* **FCBC** *problem at a cost of at most $U' = (2+\alpha)U$ where α is the approximation ratio of the color-blind clustering algorithm. The additive approximation is $\mid \mathcal{H} \mid (\epsilon + \frac{2}{L(U')})$ for the* GROUP-UTILITARIAN *objective and $\epsilon + \frac{2}{L(U')}$ for the* GROUP-EGALITARIAN *objective.*

From the theorem above, it is clear that the additive approximation guarantees we have improve when the cost does not permit small clusters. Indeed, in the absence of outlier points and for reasonable values of k, small clusters are unlikely to exist. Further, empirically we verify the smallest cluster size and find that the smallest cluster size is 159 points (see Section 2.3.4.3).

We now provide our general algorithm for fair clustering under a bounded cost **FCBC** which we denote by **ALG-FCBC**$(U,$ UNFAIRNESS-OBJECTIVE$)$ where we have made explicit reference to the dependence of **ALG-FCBC** on the given cost upper bound U and the desired UNFAIRNESS-OBJECTIVE which could either be the GROUP-UTILITARIAN,GROUP-EGALITARIAN, or GROUP-LEXIMIN objective.

ALG-FCBC$(U,$ UNFAIRNESS-OBJECTIVE$)$ (see Algorithm 3) involves two steps, in step

(1): we use a color-blind approximation algorithm to find the cluster centers S, in step **(2):** we call the algorithm $\mathbf{ALG\text{-}FABC}(S, U', \text{UNFAIRNESS-OBJECTIVE})$ for the $\mathbf{FABC}$ problem. It should be noted that we have fed $\mathbf{ALG\text{-}FABC}$ the set of centers S from step **(1)**, further the cost upper bound for $\mathbf{ALG\text{-}FABC}$ is set to $U' = (2+\alpha)U$ while the UNFAIRNESS-OBJECTIVE remains unchanged. We further note that $\mathbf{ALG\text{-}FABC}$ will have the same clustering objective as $\mathbf{ALG\text{-}FCBC}$, e.g. if $\mathbf{ALG\text{-}FCBC}$ is given the k-median objective so well $\mathbf{ALG\text{-}FABC}$.

Clearly, from algorithm $\mathbf{ALG\text{-}FCBC}$ the $\mathbf{FCBC}$ problem is closely related to the $\mathbf{FABC}$ problem. In fact, we establish the following general theorem for all clustering objectives: k-center, k-median, and k-means that shows that an algorithm which solves the $\mathbf{FABC}$ problem with provable guarantees can be used to solve the $\mathbf{FCBC}$ problem with provable guarantees:

Theorem 2.3.12. *For any clustering objective and both the* GROUP-UTILITARIAN *and* GROUP-EGALITARIAN *objectives, given an algorithm that solves fair assignment under a bounded cost* $\mathbf{FABC}$ *with additive approximation* μ*, the fair clustering under a bounded cost* $\mathbf{FCBC}$ *problem can be solved with an additive approximation of* μ *and at a cost of at most* $(2+\alpha)U$*, where* α *is the approximation ratio of the color-blind clustering algorithm.*

Proof. Let S and ϕ be the set of centers and assignment of the color-blind algorithm. Let S^* and ϕ^* be the optimal set of centers and assignment for the fair assignment under bounded cost $\mathbf{FABC}$. Let ϕ' be an assignment that routes the vertices from their center in S^* to the nearest center in S, i.e. for a given vertex j, $\phi'(j) = \arg\min_{i' \in S} d(i', \phi^*(j))$. Based on this setting we

can upper bound the objective based on the following:

$$\begin{aligned} d(j,\phi'(j)) &\leq d(j,\phi^*(j)) + d(\phi'(j),\phi^*(j)) \\ &\leq d(j,\phi^*(j)) + d(\phi(i),\phi^*(j)) \\ &\leq d(j,\phi^*(j)) + d(j,\phi^*(j)) + d(j,\phi(j)) \\ &\leq 2d(j,\phi^*(j)) + d(j,\phi(j)) \end{aligned}$$

It follows then by the triangle inequality of the p-norm and the non-negativity of the components, that $\Big(\sum_{j\in\mathcal{C}} d^p(j,\phi'(j))\Big)^{1/p} \leq 2\Big(\sum_{j\in\mathcal{C}} d^p(j,\phi^*(j))\Big)^{1/p} + \Big(\sum_{j\in\mathcal{C}} d^p(j,\phi(j))\Big)^{1/p} \leq 2\,U + \alpha U = (2+\alpha)U$. Note that in the last inequality the bounded the color-blind cost as follows: $\Big(\sum_{j\in\mathcal{C}} d^p(j,\phi(j))\Big)^{1/p} \leq \alpha\,\mathrm{OPT}_{\text{cb}} \leq \alpha U$, where as noted the optimal color-blind cost OPT_{cb} is upper bounded by U, i.e. $\mathrm{OPT}_{\text{cb}} \leq U$ otherwise the problem would not be feasible. This proves the upper bound on the objective.

Now we establish guarantees on the proportions. For a given center s in S, let $N(s) = \{i' \in S^* | s = \arg\min_{i\in S} d(i,i')\}$, i.e. $N(s)$ is the set of centers in S^* routing their vertices to s. Denote the set of points assigned to cluster i' by $\phi^{*-1}(i')$, i.e. $\phi^{*-1}(i') = \{j \in \mathcal{C}\,|\phi^*(j) = i'\}$. Then for any color h we have that:

$$\begin{aligned} &\min_{i'\in N(s)} \frac{\big(\sum_{j\in\phi^{*-1}(i'),\chi(j)=h} 1\big)}{|\phi^{*-1}(i')|} \leq \\ &\frac{\sum_{i'\in N(s)}\big(\sum_{j\in\phi^{*-1}(i'),\chi(j)=h} 1\big)}{\sum_{i'\in N(s)}|\phi^{*-1}(i')|} \leq \\ &\max_{i'\in N(s)} \frac{\big(\sum_{j\in\phi^{*-1}(i'),\chi(j)=h} 1\big)}{|\phi^{*-1}(i')|} \end{aligned}$$

That is the final color proportion will be within the lower and upper proportions of the routing centers. It follows that Δ_h does not increase for any color and that the GROUP-UTILITARIAN, GROUP-EGALITARIAN, and GROUP-LEXIMIN objectives using ϕ' are not greater than that of the optimal solution.

The above facts, combined with the premise of having an algorithm that solves the fair assignment under bounded cost **FABC** with an additive violation of μ completes the proof. □

Algorithm 3 :**ALG-FCBC**(U, UNFAIRNESS-OBJECTIVE)

1: Choose a set of centers S by running a color-blind clustering algorithm of approximation ratio α.
2: Set $U' = (2 + \alpha)U$ and call **ALG-FABC**(S, U', UNFAIRNESS-OBJECTIVE)

2.3.2.1 Fair Assignment Under a Bounded Cost

Algorithm block 4 shows the steps of our algorithm **ALG-FABC** for the **FABC** objective. In step **(1):** we search for the optimal proportional violations given the bound on the clustering cost U using LPs. Having found the near-optimal solution, in step **(2):** we round the possibly fractional solution to a feasible integer solution using a netowk flow algorithm. We note that the details of the search done in step **(1)** depend on the objective, i.e., GROUP-UTILITARIAN or GROUP-EGALITARIAN.

Algorithm 4 :**ALG-FABC**(S, U, UNFAIRNESS-OBJECTIVE)

1: Given the UNFAIRNESS-OBJECTIVE, search for the optimal proportion violation values Δ_h at a cost upper bound of U using the feasibility LPs of (2.11).
2: Apply network flow rounding to the LP solution with the optimal value.
3: **return** the set of centers S and the assignment function ϕ (resulting from the rounded LP solution).

We note that in fair assignment under a bounded cost **FABC** the set of centers S has

already been chosen and the optimization is done only over the assignment ϕ of points to centers. We let x_{ij} be a decision variable that equals 1 if point j is assigned to center $i \in S$ and 0 otherwise. Note that the values of x_{ij} are a way to represent the assignment function ϕ. Regardless of the objective that is being minimized, the following set of constraints must hold:

$$\sum_{i,j} d^p(i,j)x_{ij} \leq U^p \tag{2.11a}$$

$$\forall j \in \mathcal{C} : \sum_{i \in S} x_{ij} = 1, \quad x_{ij} \in [0,1] \tag{2.11b}$$

$$\forall h \in \mathcal{H} : \Delta_h \in [0,1] \tag{2.11c}$$

$$\forall\, h \in \mathcal{H}, \forall i \in S : (\beta_h - \Delta_h)\Big(\sum_{j \in \mathcal{C}} x_{ij}\Big) \leq \sum_{j \in \mathcal{C}^h} x_{ij} \leq (\alpha_h + \Delta_h)\Big(\sum_{j \in \mathcal{C}} x_{ij}\Big) \tag{2.11d}$$

For the k-center problem, the first constraint (2.11a) is replaced by $\forall j \in \mathcal{C} : x_{ij} = 0$ if $d(i,j) > U$. Note that in the above we have $x_{ij} \in [0,1]$ which is a relaxation of $x_{ij} \in \{0,1\}$, as the latter would result in an intractable mixed-integer program. With our variables being x_{ij} and Δ_h it is reasonable to consider a convex optimization approach. That is, we could choose to minimize the objective GROUP-UTILITARIAN or the objective GROUP-EGALITARIAN with our set of constraints being (2.11). Looking at the form of the GROUP-UTILITARIAN and the GROUP-EGALITARIAN objectives, it is not difficult to see that they are linear (and therefore convex) in the parameters x_{ij} and Δ_h, however as the following theorem shows, the constraint set (2.11) is not convex. In fact, either of the proportion bounds alone in constraint (2.11d) would lead to a non-convex set. The non-convexity of the constraint set implies that the resulting optimization problem would also be non-convex:

Theorem 2.3.13. *The constraint set (2.11) is not convex.*

Proof. The non-convexity of the constraint set (2.11) can be shown even when ignoring the upper proportionality constraint, i.e. constraint (2.11d) only with the lower bound. Specifically, we would have the following constraint set:

$$\sum_{i,j} d^p(i,j)x_{ij} \leq U^p \tag{2.12a}$$

$$\forall j \in \mathcal{C} : \sum_{i \in S} x_{ij} = 1, \quad x_{ij} \in [0,1] \tag{2.12b}$$

$$\forall h \in \mathcal{H} : \Delta_h \in [0,1] \tag{2.12c}$$

$$\forall\, h \in \mathcal{H}, \forall i \in S : (\beta_h - \Delta_h)\Big(\sum_{j \in \mathcal{C}} x_{ij}\Big) \leq \sum_{j \in \mathcal{C}^h} x_{ij} \tag{2.12d}$$

Now, assume that the upper bound on the cost U is sufficiently large (this would let assignments of a high cost remain feasible). Consider the case of two colors: red and blue, with $\beta_{\text{red}} = \beta_{\text{blue}} = \frac{1}{2}$. Let each color constitute half the dataset, i.e. $|\,\mathcal{C}^{\text{red}}\,| = |\,\mathcal{C}^{\text{blue}}\,| = \frac{n}{2}$, clearly $|\,\mathcal{C}\,| = 2|\,\mathcal{C}^{\text{red}}\,| = 2|\,\mathcal{C}^{\text{blue}}\,| = n$. Set the number of clusters to two ($k = 2$), consider the following two feasible solutions $x^1_{ij}, \Delta^1_{\text{red}}, \Delta^1_{\text{blue}}$ and $x^2_{ij}, \Delta^2_{\text{red}}, \Delta^2_{\text{blue}}$ with $\Delta^1_{\text{blue}} = \Delta^2_{\text{blue}} = 1$, then the following holds

(note that $\alpha = \frac{2}{3}$):

For $x^1_{ij}, \Delta^1_{\text{red}}$:

$$\text{cluster 1: } \sum_{j \in \mathcal{C}^{\text{red}}} x^1_{1j} = \sum_{j \in \mathcal{C}^{\text{blue}}} x^1_{1j} = \alpha \frac{n}{2} = \frac{2}{3}\frac{n}{2} = \frac{n}{3}$$

$$\text{cluster 2: } \sum_{j \in \mathcal{C}^{\text{red}}} x^1_{2j} = \sum_{j \in \mathcal{C}^{\text{blue}}} x^1_{2j} = (1-\alpha)\frac{n}{2} = \frac{1}{3}\frac{n}{2} = \frac{n}{6}$$

$$|C_2| = \sum_{j \in \mathcal{C}} x^1_{2j} = \frac{n}{3}$$

$$\Delta^1_{\text{red}} = 0$$

For $x^2_{ij}, \Delta^2_{\text{red}}$:

$$\text{cluster 1: } \sum_{j \in \mathcal{C}^{\text{red}}} x^2_{1j} = \frac{n}{2},$$

$$\sum_{j \in \mathcal{C}^{\text{blue}}} x^2_{1j} = (1 - (\alpha + \frac{1}{n/2}))\frac{n}{2} = \frac{n}{6} - 1$$

$$\text{cluster 2: } \sum_{j \in \mathcal{C}^{\text{red}}} x^2_{2j} = 0,$$

$$\sum_{j \in \mathcal{C}^{\text{blue}}} x^2_{2j} = (\alpha + \frac{1}{n/2})\frac{n}{2} = (\frac{2}{3} + \frac{1}{n/2})\frac{n}{2} = \frac{n}{3} + 1$$

$$|C_2| = \sum_{j \in \mathcal{C}} x^2_{2j} = \frac{n}{3} + 1$$

$$\Delta^2_{\text{red}} = \frac{1}{2}$$

We now form a simple convex combination of the two solutions $x_{ij} = \frac{1}{2}(x^1_{ij} + x^2_{ij}), \Delta_{\text{red}} = \frac{1}{2}(\Delta^1_{\text{red}} + \Delta^2_{\text{red}}) = \frac{1}{4}$. Constraints (2.12a), (2.12b), and (2.12c) would clearly be satisfied, but if

we consider constraint (2.12d) for the red color and the second cluster, then we have:

$$RHS = \sum_{j \in \mathcal{C}^{\text{red}}} x_{2j} = \frac{n}{12}$$
$$LHS = (\frac{1}{2} - \frac{1}{4})(\frac{n}{3} + \frac{1}{2}) = \frac{n}{12} + \frac{1}{8}$$

It is clear that $LHS \leq RHS$ does not hold and therefore, the constraint is not satisfied for the convex combination and therefore the constraint set of the problem is indeed not convex.

A similar assignment of solutions can be used to show that the set is not convex if we were to consider only the over-representation constraint in (2.11d) instead. □

Although the constraint set (2.11) is not convex, if we fix the values of Δ_h then we clearly have a simple feasibility LP with variables x_{ij}. We therefore take an approach where for a given objective (GROUP-UTILITARIAN or GROUP-EGALITARIAN), we search for the corresponding optimal values of Δ_h by running the feasibility LP of (2.11). Note that with a given set of values for Δ_h, we can obtain the corresponding value for the GROUP-UTILITARIAN or GROUP-EGALITARIAN objectives and therefore the LP does not need an objective: a feasibility check suffices. Further, since we only use non-trivial values for $\Delta_h \in [0, 1]$, constraint (2.11c) can be omitted. Below we discuss how we use the feasibility LPs of (2.11) to obtain LP solutions that are approximately optimal (having bounded additive approximation from the optimal) for the GROUP-UTILITARIAN and GROUP-EGALITARIAN objectives, respectively. Since these resulting LP solutions could contain fractional values, i.e., it is possible to have a value $x_{ij} \notin \{0, 1\}$, the approximately optimal LP solution would have to be rounded to an integral solution. This rounding further degrades the approximation, but we show that this degradation is not large and

can be bounded. The details of the rounding scheme are shown below as well. The search algorithm (for the GROUP-UTILITARIAN or GROUP-EGALITARIAN objective), followed by the rounding scheme, lead to an algorithm for **FABC**.

Search Algorithm for GROUP-EGALITARIAN and GROUP-LEXIMIN Objectives: The first step we take is to discretize the space by a parameter $\epsilon \in (0, 1)$. For convenience, we set $\epsilon = \frac{1}{r}$ where $r \in \mathbb{Z}^+$, i.e., r is a positive integer. Accordingly, instead of interacting with the continuous interval $[0, 1]$ for the proportional violations, we instead interact with $E_\epsilon = \{\epsilon, 2\epsilon, \ldots, \ldots, (\frac{1}{\epsilon} - 1)\epsilon, 1\}$, with $| E_\epsilon | = \frac{1}{\epsilon}$. For all colors, their violation Δ_h is set to the same value and the optimal solution is found simply by doing binary search over the set E_ϵ by running the feasibility LP (2.11).

Theorem 2.3.14. *For* **FABC** *with the* GROUP-EGALITARIAN *objective, we can use* $O\Big(\log\Big(\frac{1}{\epsilon}\Big)\Big)$*–many LP runs to get a solution with an additive approximation of* ϵ*.*

Proof. The proof follows directly by using binary search. □

We provide a heuristic algorithm for the GROUP-LEXIMIN objective; a rough sketch follows. In the first step, it obtains the GROUP-EGALITARIAN solution. Then, it proceeds by finding a color that cannot improve beyond the current optimal violation; if more than one color is found, then one of these colors is randomly picked. The algorithm then looks for the optimal violation for the remaining colors, having the violations of the previous colors fixed. These steps are followed until no color can have its proportional violation improved.

Search Algorithm for the GROUP-UTILITARIAN Objective: We follow the same discretization step as for the GROUP-EGALITARIAN objective. We describe our algorithm for the important two-color case with *symmetric* upper and lower bounds we show a search algorithm

that requires only $O\left(\log \frac{1}{\epsilon}\right)$ LP runs. The two color case with *symmetric* upper and lower bounds is that where the two colors h_1 and h_2 are present with proportions r_1 and r_2 in the dataset, and the proportion bounds are set to $\alpha_i = r_i + \lambda_i, \beta_i = r_i - \lambda_i$ for $i \in \{1, 2\}$ and some valid $\lambda_1, \lambda_2 \in [0, 1]$. The key observation for the two-color symmetric case is that the proportion of one color implies the proportion of the other; hence, we can run binary search over the set E_ϵ.

Theorem 2.3.15. *For* **FABC** *with two colors, symmetric lower & upper bounds, and the* GROUP-UTILITARIAN *objective, we can use* $O\left(\log(\frac{1}{\epsilon})\right)$*–many LP runs to get a solution with an additive approximation of* $|\mathcal{H}|\epsilon = 2\epsilon$.

Proof. We first point out the following definition and observations. For the two color case, our color set is $\mathcal{H} = \{h_1, h_2\}$. Further, we denote the proportions for color i by r_i where $r_i = \frac{|\{j | j \in \mathcal{C}, \chi(j) = i\}|}{|\mathcal{C}|} = \frac{|\mathcal{C}^i|}{|\mathcal{C}|}$. We use color h_1 to denote the color with less points, i.e. $r_1 \leq r_2$. The upper and lower bounds we consider for each color are: $\beta_i = (1 - \delta)r_i$ and $\alpha_i = (1 + \delta)r_i$. The idea behind the algorithm is that the proportions of one color imply the proportion of the other color.

Algorithm for **FABC** with *two colors* and *symmetric lower and upper proportion bounds*: Our algorithm is based on the simple observation shown in figure 2.9

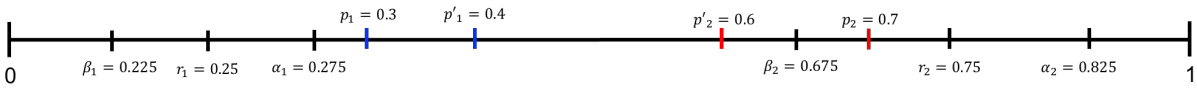


Figure 2.9: Proportions and bounds for two colors. $r_1 = 0.25, r_2 = 0.75$, $\lambda_i = \delta r_i$ for $i \in \{1, 2\}$ where $\delta = 0.1$. Notice how if color 1 violates the upper bound by having $p_1 = 0.3$, then we must have $p_2 = 0.7$, but color 2 is not violating. On the other hand, a violation for color 1 with $p'_1 = 0.4$ implies $p'_2 = 0.6$ which causes a violation for color 2.

Without loss of generality, let $\lambda_1 \leq \lambda_2$, based on the observation in figure 2.9, we have the following claim:

Claim 2.3.16. *If* $\Delta_1 < \lambda_2 - \lambda_1$, *then* $\Delta_2 = 0$. *If* $\Delta_1 \geq \lambda_2 - \lambda_1$, *then* $\Delta_2 = \Delta_1 - (\lambda_2 - \lambda_1)$

Proof. Let color 1 have a Δ_1 violating proportion of p_1, then in some cluster $p_1 = \alpha_1 + \Delta_1$ or $p_1 = \beta_1 - \Delta_1$.

Consider the case where $p_1 = \alpha_1 + \Delta_1$, then $p_2 = 1 - p_1 = 1 - \alpha_1 - \Delta_1$. Now if $\Delta_1 < \lambda_2 - \lambda_1$, then we have $p_2 > 1 - \alpha_1 - (\lambda_2 - \lambda_1) = 1 - \lambda_2 + \lambda_1 - \alpha = (1 - r_1) - \lambda_2 = r_2 - \lambda_2 = \beta_2$ this means that color 2 does not violate the lower bound. If we assume that color 2 violates the upper bound by an amount $\Delta_2 > 0$, then this would imply that $p_1 = 1 - p_2$ and the lower violation for color 1 would be $\beta_1 - p_1 = \beta_1 - (1 - \alpha_2 - \Delta_2) = \beta_1 - 1 + \alpha_2 + \Delta_2 = r_1 - \lambda_1 + r_2 + \lambda_2 + \Delta_2 - 1 = 1 + (\lambda_2 - \lambda_1) + \Delta_2 - 1 = (\lambda_2 - \lambda_1) + \Delta_2 > (\lambda_2 - \lambda_1)$ which is a contradiction since we assumed that $\Delta_1 < (\lambda_2 - \lambda_1)$.

Similarly, if $\Delta_1 \geq (\lambda_2 - \lambda_1)$, then we have $\Delta_2 = \beta_2 - (1 - \alpha_1 - \Delta_1) = \beta_2 - 1 + \alpha_1 + \Delta_1 = r_2 - \lambda_2 + r_1 + \lambda_1 - 1 + \Delta_1 = \lambda_1 - \lambda_2 + \Delta_1 = \Delta_1 - (\lambda_2 - \lambda_1)$. Now if we assume that color 2 has a violation of the upper bound by an amount $\Delta_2' > \Delta_1 - (\lambda_2 - \lambda_1)$, this would imply that color 1 violates the lower bound by $\beta_1 - p_1 = r_1 - \lambda_1 + r_2 + \lambda_2 - 1 + \Delta_2 = (\lambda_2 - \lambda_1) + \Delta_2$ which is a contradiction since $\Delta_1 < \Delta_2 + (\lambda_2 - \lambda_1)$, therefore color 2 cannot violate by more than $\Delta_1 - (\lambda_2 - \lambda_1)$.

The case of $p_1 = \beta_1 - \Delta_1$ follows similar arguments. □

The above observations lead to algorithm 5.

Now we are ready to prove the theorem. From Claim (2.3.16) we can do binary search over the set E_ϵ using Δ_1 as done in algorithm (5). Clearly, at most $O\Big(\log(\frac{1}{\epsilon})\Big)$ many LPs will be run

Algorithm 5 GROUP-UTILITARIAN Algorithm for Two Colors with Symmetric Bounds for the GROUP-UTILITARIAN Objective

Input: set of points $\mathcal{C}$, cost upper bound U, for each color $h \in \mathcal{H}$ lower and upper proportion values β_h, α_h, error parameter ϵ.
Define the set $E_\epsilon = \{0, \epsilon, \ldots, (\frac{1}{\epsilon} - 1)\epsilon\}$
Binary search Δ_1 over the set E_ϵ by running the LP (2.11) (if $\Delta_1 < \delta(r_2 - r_1)$ then $\Delta_2 = 0$, otherwise set $\Delta_2 = \Delta_1 - \delta(r_2 - r_1)$).
return return the LP solution with the minimum Δ_1 value.

because of binary search. Further, we know that we will find a solution at most ϵ greater, i.e. we worst case best LP value is: $\Delta_1^* + \epsilon, \Delta_2^* + \epsilon = (\Delta_1^* + \Delta_2^*) + 2\epsilon = \text{OPT} + 2\epsilon$. □

For the general multi-color case, we show a search algorithm of $O((\frac{1}{\epsilon})^{|\mathcal{H}|-1})$ many LP feasibility calls, thus improving marginally over the brute force of $(\frac{1}{\epsilon})^{|\mathcal{H}|}$ many calls, see [17] for the details.

Theorem 2.3.17. *For* **FABC** *with* GROUP-UTILITARIAN *objective, we can use* $O\Big(\big(\frac{1}{\epsilon}\big)^{|\mathcal{H}|-1}\Big)$*– many LP runs to obtain an LP solution with additive approximation* $|\mathcal{H}|\epsilon$.

The Rounding Scheme and ALG-FABC Guarantees: Having obtained the optimal LP solutions for either the GROUP-UTILITARIAN or GROUP-EGALITARIAN objectives, we now round the solutions to integral values at a bounded increase to the additive approximation. To do the rounding, we apply the network flow method of [1], although other rounding methods are applicable. Given the LP solution x_{ij} and its associated proportional violations Δ_h, if we denote the rounded integral solution by $\bar{x}_{ij}$ and $\bar{\Delta}_h$, then network-flow rounding guarantees the following: **(i)** $\sum_{i,j} d^p(i,j)\bar{x}_{ij} \leq \sum_{i,j} d^p(i,j)x_{ij}$. **(ii)** $\forall i \in [k] : \Big\lfloor \sum_{j \in \mathcal{C}} x_{ij} \Big\rfloor \leq \sum_{j \in \mathcal{C}} \bar{x}_{ij} \leq \Big\lceil \sum_{j \in \mathcal{C}} x_{ij} \Big\rceil$. **(iii)** $\forall h \in \mathcal{H}, \forall i \in [k] : \Big\lfloor \sum_{j \in \mathcal{C}^h} x_{ij} \Big\rfloor \leq \sum_{j \in \mathcal{C}^h} \bar{x}_{ij} \leq \Big\lceil \sum_{j \in \mathcal{C}^h} x_{ij} \Big\rceil$.

Property **(i)** ensures that the clustering objective will not increase beyond the LP value, and thus, provided the LP cost does not exceed the upper bound on the cost U, the cost of the rounded

assignment will not exceed U as well. Property **(ii)** guarantees that the total number of points assigned to a cluster will not vary by more than 1 point. Property **(iii)** guarantees that the total number of points of a given color assigned to a cluster will not vary by more than 1 point. We can use the above properties along with with the lower bound on the size of any cluster $L(U)$ to establish the following theorem:

Theorem 2.3.18. *For the* $\mathbf{FABC}$ *problem, the rounded solution has cost of at most* U *and an additive approximation of: (1)* $|\mathcal{H}|(\epsilon + \frac{2}{L(U)})$ *for the* GROUP-UTILITARIAN *objective and (2)* $\epsilon + \frac{2}{L(U)}$ *for the* GROUP-EGALITARIAN *objective.*

Proof. First we start with the following claim:

Claim 2.3.19. $\bar{\Delta}_h < \Delta_h + \frac{2}{L(U)}$*, i.e. rounding will increase the violation by at most* $\frac{2}{L(U)}$*.*

Proof. Based on properties (ii) and (iii) from network flow rounding (mentioned above), we can

get the following bound for the upper proportion:

$$
\begin{aligned}
&\sum_{j\in\mathcal{C}^h} \bar{x}_{ij} \leq \left\lceil \sum_{j\in\mathcal{C}^h} x_{ij} \right\rceil && \text{(by property (iii))}\\
&\leq \left\lceil \min\big((\alpha_h+\Delta_h),1\big)\Big(\sum_{j\in\mathcal{C}} x_{ij}\Big) \right\rceil && \text{(problem constraint)}\\
&< \min\big((\alpha_h+\Delta_h),1\big)\Big(\sum_{j\in\mathcal{C}} x_{ij}\Big) + 1 && \text{(ceiling upper bound)}\\
&\leq \min\big((\alpha_h+\Delta_h),1\big)\Big(\sum_{j\in\mathcal{C}} \bar{x}_{ij} + 1\Big) + 1 && \text{(by property (ii))}\\
&\leq \min\big((\alpha_h+\Delta_h),1\big)\Big(\sum_{j\in\mathcal{C}} \bar{x}_{ij}\Big) + \min\big((\alpha_h+\Delta_h),1\big) + 1 &&\\
&\leq \min\big((\alpha_h+\Delta_h),1\big)\Big(\sum_{j\in\mathcal{C}} \bar{x}_{ij}\Big) + 2 && \text{(since } \min\big((\alpha_h+\Delta_h),1\big) \leq 1)\\
&\leq (\alpha_h+\Delta_h)\Big(\sum_{j\in\mathcal{C}} \bar{x}_{ij}\Big) + 2 &&
\end{aligned}
$$

This implies that the new violation for the rounded solution $\bar{\Delta}_h$ satisfies:

$$\alpha_h + \bar{\Delta}_h = \frac{\sum_{j\in\mathcal{C}^h} \bar{x}_{ij}}{\sum_{j\in\mathcal{C}} \bar{x}_{ij}} < \alpha_h + \Delta_h + \frac{2}{\sum_{j\in\mathcal{C}} \bar{x}_{ij}} \leq \alpha_h + \Delta_h + \frac{2}{L(U)}$$

Therefore, we have:

$$\bar{\Delta}_h - \Delta_h < \frac{2}{L(U)}$$

For the lower proportions, we also have:

$$
\begin{aligned}
&\sum_{j\in\mathcal{C}^h} \bar{x}_{ij} \geq \left\lfloor \sum_{j\in\mathcal{C}^h} x_{ij} \right\rfloor && \text{(by property (iii))}\\
&\geq \left\lfloor \max\big((\beta_h - \Delta_h), 0\big)\Big(\sum_{j\in\mathcal{C}} x_{ij}\Big) \right\rfloor && \text{(problem constraint)}\\
&> \max\big((\beta_h - \Delta_h), 0\big)\Big(\sum_{j\in\mathcal{C}} x_{ij}\Big) - 1 && \text{(ceiling upper bound)}\\
&\geq \max\big((\beta_h - \Delta_h), 0\big)\Big(\sum_{j\in\mathcal{C}} \bar{x}_{ij} - 1\Big) - 1 && \text{(by property (ii))}\\
&\geq \max\big((\beta_h - \Delta_h), 0\big)\Big(\sum_{j\in\mathcal{C}} \bar{x}_{ij}\Big) - \max\big((\beta_h - \Delta_h), 0\big) - 1 &&\\
&\geq \max\big((\beta_h - \Delta_h), 0\big)\Big(\sum_{j\in\mathcal{C}} \bar{x}_{ij}\Big) - 2 && \text{(since } \max\big((\beta_h - \Delta_h), 0\big) \leq 1\text{)}\\
&\geq (\beta_h - \Delta_h)\Big(\sum_{j\in\mathcal{C}} \bar{x}_{ij}\Big) - 2 &&
\end{aligned}
$$

$$
\beta_h - \bar{\Delta}_h = \frac{\sum_{j\in\mathcal{C}^h} \bar{x}_{ij}}{\sum_{j\in\mathcal{C}} \bar{x}_{ij}} > \beta_h - \Delta_h - \frac{2}{\sum_{j\in\mathcal{C}} \bar{x}_{ij}} \geq \beta_h - \Delta_h - \frac{2}{L(U)}
$$

Therefore, we have:

$$
\bar{\Delta}_h - \Delta_h < \frac{2}{L(U)}
$$

□

Now we are ready to prove the theorem. (1) For the GROUP-UTILITARIAN, by Theorems (2.3.17) and (2.3.15) the LP solution has a violation of $|\,\mathcal{H}\,| + \epsilon$, then by Claim (2.3.19) and the

definition of the GROUP-UTILITARIAN $= \sum_{h \in \mathcal{H}} \Delta_h$, the violation is at most $|\mathcal{H}|(\epsilon + \frac{2}{L(U)})$.

(2) For the GROUP-EGALITARIAN, by Theorem (2.3.14) and Claim (2.3.19), the rounded solution would have a worst case violation of $\epsilon + \frac{2}{L(U)}$ across the colors. □

Recalling the additive approximation lower bounds of Theorem 2.3.6 for the **FABC** problem, we see that we obtain a solution for **FABC** of cost at most U with near-optimal additive approximation. Specifically, our additive approximations for the GROUP-UTILITARIAN and GROUP-EGALITARIAN are $\frac{2|\mathcal{H}|}{L(U)}$ and $\frac{2}{L(U)}$ compared to their lower bounds of $\frac{|\mathcal{H}|}{8L(U)}$ and $\frac{1}{8L(U)}$, respectively.

2.3.3 Fairness Across the Clusters is not Possible

It is tempting to modify both the GROUP-UTILITARIAN and GROUP-EGALITARIAN (or GROUP-LEXIMIN) objectives to sum across the clusters instead of taking the maximum violation across the clusters. More specifically, we can replace the objectives by the following: GROUP-UTILITARIAN-SUM, which equals $\sum_{h \in \mathcal{H}, i \in [k]} \Delta_h^i$, and GROUP-EGALITARIAN-SUM, which equals $\min \max_{h \in \mathcal{H}} \sum_{i \in [k]} \Delta_h^i$, where Δ_h^i is the violation of color h in cluster i; clearly the previously-considered violation Δ_h is $\max_{i \in [k]} \Delta_h^i$. It can be seen that such an objective is more flexible. For example, the maximum violations might occur in a cluster that cannot be improved within the given bound on the clustering cost, while it may be possible to improve it for other clusters. The original GROUP-UTILITARIAN and GROUP-EGALITARIAN objectives may bring no improvement in such a situation but their above modifications could. We prove a negative result. Specifically, while we were able to approximate **FABC** by small additive values for the original objectives (Theorem 2.3.18), for the new objectives we cannot efficiently approximate

the **FABC** problems within even relatively-large additive approximations:

Theorem 2.3.20. *For* **FABC***, the objectives* GROUP-UTILITARIAN-SUM *and* GROUP-EGALITARIAN-SUM *that sum across the clusters cannot be approximated in polynomial time to within an additive approximation of* $O(n^{\delta})$ *where* δ *is a constant in* $[0, 1)$*, unless* $P = NP$*.*

Proof. We first introduce the following lemma:

Lemma 2.3.21. *Any polynomial time approximation algorithm for* **FABC** *for a general upper bound* U *must have* $\mu > 0$*, i.e. it must have a strictly greater than zero additive approximation guarantee.*

Proof. The proof follows from the proof of Theorem (2.3.1). Specifically, the proof of Theorem (2.3.1) shows that hard instances for **FABC** could have an optimal value of 0 for the GROUP-UTILITARIAN, GROUP-EGALITARIAN, and GROUP-LEXIMIN objectives, specifically when $U = \mathrm{OPT}_{\mathrm{FC}}$ where $\mathrm{OPT}_{\mathrm{FC}}$ is the optimal value of fair clustering. Therefore, if a polynomial time approximation algorithm with approximation ratio $\rho \geq 1$ and additive approximation $\mu \geq 0$ is ran over such hard instances, then it would output a solution of value $\rho\,\mathrm{OPT} + \mu = \rho(0) + \mu = \mu$. If the algorithm has $\mu = 0$, then it would mean that the problem has been solved optimally which is impossible unless $P = NP$. Therefore, $\mu > 0$. □

By the result of the above lemma we know that we can hard instances with $\mathrm{OPT} = 0$ and that any polynomial time algorithm should have an additive approximation $\mu > 0$. Further, we consider the same X3C reduction of Theorem 2.3.1 and Figure 2.8 for **FABC** with the centers set to the points of $\mathcal{F}$.

To prove the theorem, suppose by contradiction that an algorithm $\mathcal{A}$ exists that guarantees an additive approximation of $O(n^{\delta})$ for $\delta \in [0, 1)$. Suppose, we are given an instance of the

problem with optimal solution value of OPT and n many points. Note by Lemma 2.3.10 if $\Delta^i_{\text{red}} < \frac{1}{|C_i|}$, then there us no violation. It follows that if $\sum_{i\in[k]}(\Delta^i_{\text{red}} + \Delta^i_{\text{blue}}) < \frac{1}{4n}$ then we have no violation.

Now, create D many duplicates of the given set of points. Let the distance between the points belonging to the same duplicate be the same as in the original instance, whereas for points in different duplicates the distance is infinity. Further, let the number of centers be Dk where each duplicate has k many centers assigned at the same points as the original instance. Given the original upper bound on the clustering objective U, the new upper bound U' is set to $U' = U$ for the k-center, $U' = DU$ for the k-median, and $U' = \sqrt{D}U$ for the k-means objectives.

If this modified instance is given to $\mathcal{A}$, then the output would have a value of at most $\rho D\,\mathrm{OPT} + c(Dn)^\delta$ for some $c > 0$. If $D > \frac{1}{4^\delta - 1} c^{\frac{1}{1-\delta}} n^{\frac{1+\delta}{1-\delta}}$ (which is polynomial in n), then the average violation across the duplicates is:

$$\begin{aligned}
&\frac{\rho D\,\mathrm{OPT} + c(Dn)^\delta}{D} = \rho\,\mathrm{OPT} + cn^\delta D^{\delta-1} \\
&< \rho\,\mathrm{OPT} + c\frac{1}{4}n^\delta c^{\frac{\delta-1}{1-\delta}} n^{\frac{(1+\delta)(\delta-1)}{1-\delta}} = \rho\,\mathrm{OPT} + \frac{1}{4n} = 0 + \frac{1}{4n}
\end{aligned}$$

This means that there must exist at least one duplicate for which the violation is at most $\frac{1}{4n}$ which means that the problem has been exactly in polynomial time which is impossible unless $P = NP$. □

2.3.4 Experiments

We validate our algorithms on datasets from the UCI repository [54]. The results here are for k-means clustering.

2.3.4.1 GROUP-UTILITARIAN Experiments

We use the **Adult** and **Census1990** datasets with self-reported gender (male or female) as the attribute. We note that both datasets explicitly use categorical labels for this socially-complex concept, and acknowledge that this is reductive [55]. Figure 2.10 shows the PoF versus the achieved GROUP-UTILITARIAN objective, with $\delta = 0.1$. As expected, as the price of fairness increases (higher bound on the cost), we can further minimize the proportional violations. Eventually the GROUP-UTILITARIAN objective becomes less than 0.1 and even very close to zero. We also observe that at a given cost upper bound, we can achieve lower values for the GROUP-UTILITARIAN objective when the number of clusters (k) is lower.

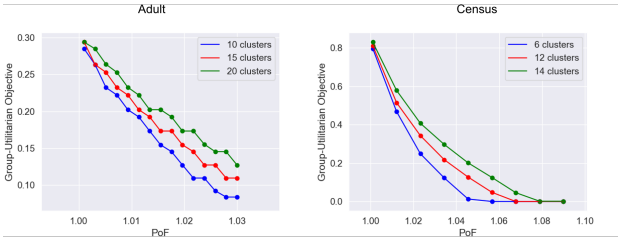


Figure 2.10: PoF vs the GROUP-UTILITARIAN objective for the **Adult** and **Census1990** datasets.

2.3.4.2 GROUP-EGALITARIAN and GROUP-LEXIMIN Experiments

We again use the **Adult** and **Census1990** datasets. However, for **Adult**, we set the fairness attribute to race which—in this dataset, and with the same inherent social caveats as the categorization of gender—has 5 groups (colors). For **Census1990**, we set the fairness attribute to age where we have three age groups.[6] We set $\delta = 0.05$ and $k = 10$ for **Adult** and $\delta = 0.1$ and $k = 5$ for **Census1990**. Figure 2.11 shows the results of our algorithm. We notice that for some colors

[6] **Census1990** actually has 8 age groups. For better interpretability of the results, we merge nearby groups $\{0, 1, 2\}$, $\{3, 4, 5\}$, and $\{6, 7, 8\}$ to form 3 groups.

smaller violations are harder to achieve and we need to set the maximum allowable clustering cost to larger values to reduce their violations.

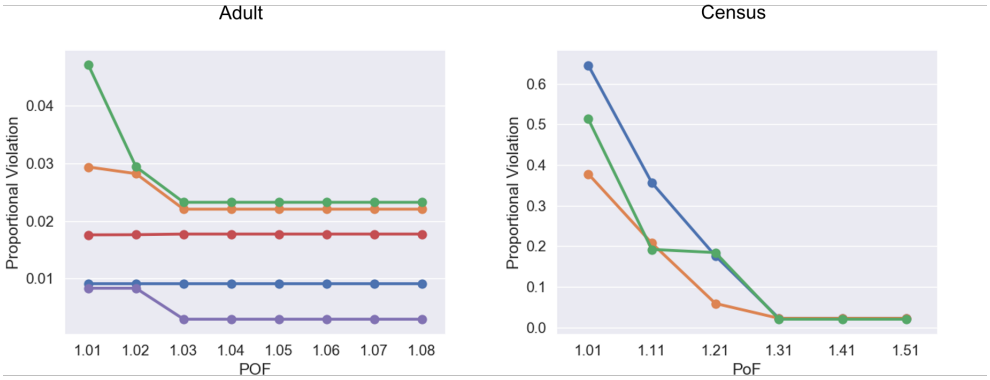


Figure 2.11: PoF versus the proportional violation for different groups (each colored graph is a group) in the **Adult** and **Census1990** datasets.

2.3.4.3 Checking the Size of the Smallest Cluster

As mentioned in Theorem 2.3.11 our approximations are dependent on the size of the smallest cluster in the solution. While it is not tractable to obtain the value of $L(U)$ for a given U, we can still empirically check the size of the smallest cluster in the cost bounded clusterings we obtain. We note that, throughout, we do not impose any lower bound on the cluster size in our algorithm. For the above experiments we considered, we find that the minimum cluster size (across all choices of k) are as follows: **Adult** (159 points), **Census1990** (171 points). The fact that the size of the smallest cluster is large means that we are achieving small (accurate) additive approximations with near-optimal objective values and when we obtain a large objective value it is because of how stringent the cost upper bound is.

2.4 Fair Labeled Clustering

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 [56–59]. 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 [60–62] 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 represen-

tation 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 [63] in which we have a dataset of candidates,[7] 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 examination (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 [64, 65]. 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

[7] 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`.

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) [51].

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.

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 [56, 58, 63] or in the case of the unassigned labels we would have a fixed set of centers. Further, the work of [66] 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 [1, 17].[8] 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.

[8] 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.

2.4.1 Further Definitions for the Labeled Fair Clustering Problem

The cardinality of the set of colors is R, i.e. $|\mathcal{H}| = R$. We refer to the set of points with color $h \in \mathcal{H}$ by $\mathcal{C}^h$. 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 $\ell : S \to \mathcal{L}$ assigns centers to labels. Our problem always involves finding an assignment from points to centers, $\phi : \mathcal{C} \to S$ 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 the k-center, k-median, and k-means objectives. We consider the number of colors R 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 $|\mathcal{L}| = m > 2$.

2.4.1.1 Labeled Clustering with Assigned Labels (LCAL):

In this problem the labels of the centers have been assigned, i.e. the function ℓ is fully known and fixed. We look for an assignment ϕ which is the optimal solution to the following

problem:

$$\min_{\phi} \Big(\sum_{j \in \mathcal{C}} d^p(j, \phi(j)) \Big)^{1/p} \tag{2.13a}$$

$$\forall L \in \mathcal{L}, \forall h \in \mathcal{H} : l_h^L \sum_{\substack{i \in S \\ \ell(i)=L}} |\, \mathcal{C}_i \,| \leq \sum_{\substack{i \in S \\ \ell(i)=L}} |\, \mathcal{C}_i^h \,| \leq u_h^L \sum_{\substack{i \in S \\ \ell(i)=L}} |\, \mathcal{C}_i \,| \tag{2.13b}$$

$$\forall L \in \mathcal{L} : (LB)_L \leq \sum_{i \in S : \ell(i)=L} |\, \mathcal{C}_i \,| \leq (UB)_L \tag{2.13c}$$

where $\mathcal{C}_i$ refers to the points ϕ assigns to the center i, i.e. $\mathcal{C}_i = \{j \in \mathcal{C} \,|\phi(j) = i\}$. $\mathcal{C}_i^h = \mathcal{C}_i \cap \mathcal{C}^h$, i.e. the subset of $\mathcal{C}_i$ with color h. l_h^L and u_h^L are lower and upper proportional bounds for color h. Clearly, $l_h^L, u_h^L \in [0, 1]$. Constraints (2.13b) are the proportionality (fairness) constraints that are to be satisfied in fair labeled clustering. Notice how we have a superscript L in l_h^L and u_h^L, 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 $l_h^N = 0$ and $u_h^N = 1$ but we may want to enforce lower representation bounds in the positive label and therefore set l_h^P 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 (2.13c), $(LB)_L$ 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.

2.4.1.2 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:

$$\min_{\phi,\ell} \Big(\sum_{j \in \mathcal{C}} d^p(j, \phi(j)) \Big)^{1/p} \tag{2.14a}$$

$$\forall L \in \mathcal{L}, \forall h \in \mathcal{H} : l_h^L \sum_{\substack{i \in S \\ \ell(i)=L}} |\, \mathcal{C}_i \,| \leq \sum_{\substack{i \in S \\ \ell(i)=L}} |\, \mathcal{C}_i^h \,| \leq u_h^L \sum_{\substack{i \in S \\ \ell(i)=L}} |\, \mathcal{C}_i \,| \tag{2.14b}$$

$$\forall L \in \mathcal{L} : (LB)_L \leq \sum_{i \in S : \ell(i)=L} |\, \mathcal{C}_i \,| \leq (UB)_L \tag{2.14c}$$

$$\forall L \in \mathcal{L} : (CL)_L \leq |S^L| \leq (CU)_L \tag{2.14d}$$

Note how in the above objective ℓ has been added as an optimization variable unlike the objective in (2.13) for LCAL. Further, we have added constraint (2.14d) where S^L refers to the subset of centers that have been assigned label L by the function ℓ, i.e. $S^L = \{i \in S | \ell(i) = L\}$. 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.

2.4.2 Algorithms and Theoretical Guarantees for LCAL

2.4.2.1 LCAL is Polynomial Time Solvable:

LCAL is problem (2.13) where we have a collection of centers and we wish to minimize a clustering objective subject to proportionality constraints (2.13b) and possible constraints on the number of points each label is assigned (2.13c). Fair assignment[9] 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:

$$\min_{\phi} \Big(\sum_{j \in \mathcal{C}} d^p(j, \phi(j)) \Big)^{1/p} \tag{2.15a}$$

[9]Fair assignment [1, 14, 40] is a sub-problem solved in fair clustering to finally yield a full algorithm for fair clustering.

$$\forall i \in S, \forall h \in \mathcal{H} : l_h |\mathcal{C}_i| \leq |\mathcal{C}_i^h| \leq u_h |\mathcal{C}_i| \tag{2.15b}$$

It may be thought that the above optimization is simpler than that of $LCAL$ (2.13), since all clusters have to satisfy the same proportionality bounds and there is no bound on the total number of points assigned to a any specific cluster. However, [1, 17] 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 2.4.1. *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. $(n_1, \ldots, n_L, \ldots, n_m)$ 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 $V = \{s\} \cup \mathcal{C} \cup (\cup_{h \in \mathcal{H}} S^h) \cup (\cup_{h \in \mathcal{H}} \mathcal{L}^h) \cup \mathcal{L} \cup \{t\}$. Vertex s is the source, further we have a vertex for each point, hence the set of vertices $\mathcal{C}$. For each color $h \in \mathcal{H}$ we create a vertex for each center in S and for each label in $\mathcal{L}$, these vertices constitute the sets $\cup_{h \in \mathcal{H}} S^h$ and $\cup_{h \in \mathcal{H}} \mathcal{L}^h$, respectively. We also have a vertex for each label in $\mathcal{L}$ and finally the sink t.

- **Edges:** the set of edges is $E = E_{s \to \mathcal{C}} \cup E_{\mathcal{C} \to S^h} \cup E_{S^h \to \mathcal{L}^h} \cup E_{\mathcal{L}^h \to \mathcal{L}} \cup E_{\mathcal{L} \to t}$. $E_{s \to \mathcal{C}}$ consists of edges from the source s to every point $j \in \mathcal{C}$, $E_{\mathcal{C} \to S^h}$ consists of edges from every point $j \in \mathcal{C}$ to the center of vertices of the same color in S^h, $E_{S^h \to \mathcal{L}^h}$ consists of edges from the colored centers to their corresponding label of the same color, $E_{\mathcal{L}^h \to \mathcal{L}}$ 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 $E_{s \to \mathcal{C}}$ have a capacity of 1, the edges of $E_{\mathcal{L}^h \to \mathcal{L}}$ have a capacity of $\lfloor u_h^L n_L \rfloor$, the edges of $E_{\mathcal{L} \to t}$ have a capacity of n_L.

- **Demands:** the vertices of $\mathcal{L}^h$ have a demand of $\lceil l_h^L n_L \rceil$, the vertices of $\mathcal{L}$ have a demand of n_L.

- **Costs:** all edges have a cost of zero except the edges of $E_{\mathcal{C} \to S^h}$ 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, $\forall (j, i^{\chi(j)}) \in E_{\mathcal{C} \to S^h}$, $\text{cost}(j, i^{\chi(j)}) = d^p(j, i)$ where $p = 1$ for the k-median and $p = 2$ for the k-means.

We write the cost for a constructed flow graph as $\sum_{j \in \mathcal{C}, i \in S} d^p(j, i) x_{ij}$ 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 $E_{\mathcal{C} \to S^h}$, we instead pick a value d from the set of possible distances $d(j, i)$ where $j \in \mathcal{C}, i \in S$ 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. □

2.4.2.2 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. First, we note that each

color $h \in \mathcal{H}$ exists in proportion $r_h = \frac{|\mathcal{C}^h|}{|\mathcal{C}|}$ where we refer to r_h as the population proportion. The case of exact population proportions for the positive and negative labels, is the one where $\forall\, h \in \mathcal{H}, \forall L \in \{P, N\} : l_h^L = u_h^L = r_h = \frac{|\mathcal{C}^h|}{|\mathcal{C}|}$

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 $= 3c$, 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_{atomic}^h.

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 (6)). In the first step we start with all points in N, then in step 2 we move the minimum number of n_{atomic}^h for each color h to satisfy the size bounds for each label (constraint (2.13c)). 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 6 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** $i = \left(\frac{n}{\sum_{h\in\mathcal{H}} q_h}\right)$ 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 2.4.2. *Algorithm (6) 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 (2.13b) 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 (2.13b) 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 $O\big(n(\log n + k)\big)$. □

The above algorithms can be generalized to give all solution values for arbitrary choices of label size bounds (constraint(2.13c)) 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).

2.4.3 Algorithms and Theoretical Guarantees for LCUL

2.4.3.1 Computational Hardness of LCUL

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) [67] 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. See figure 2.12 for an example. This is essentially a reduction we follow in all proofs, sometimes changes are introduced and mentioned explicitly in the proofs.

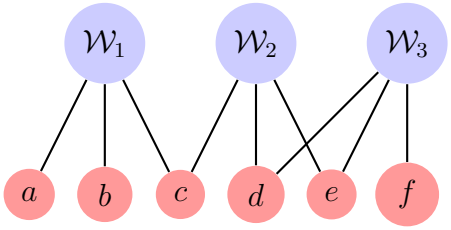


Figure 2.12: Example of the reduction for theorem (2.4.3). This is an instance of the LUCL problem for an instance $U = \{a, b, c, d, e, f\}$, $\mathcal{W}_1 = \{a, b, c\}$, $\mathcal{W}_2 = \{c, d, e\}$ and $\mathcal{W}_3 = \{d, e, f\}$ with $q = 2$, $|U| = 3q$ and $t = 3$.

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 (2.14)) we would still have an NP-hard problem.

Theorem 2.4.3. *For the* LCUL *problem with two labels and two colors, dropping one of the constraints(2.14b), (2.14c), or (2.14d) still leads to an NP-hard problem.*

Proof. We begin with the following lemma:

Lemma 2.4.4. *Even if the color-proportionality constraint (2.14b) are ignored*[10] 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

[10] We can simply remove the constraint or set $l_h^L = 0, u_h^L = 1, \forall\, h \in \mathcal{H}, L \in \mathcal{L}$.

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.

□

Now we instead ignore the constraints on the number of points a label should receive, i.e. constraints (2.14c) and keep the proportionality constraints. We show that this also results in an NP-hard problem as demonstrated in the theorem below:

Lemma 2.4.5. *Even if we do not specify the number of points a label should receive (constraint(2.14c)),* LCUL *is NP-hard.*

Proof. Similar to the proof of theorem (2.4.4) 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. $l^N_{\text{red}} = l^N_{\text{blue}} = 0$ and $u^N_{\text{red}} = u^N_{\text{blue}} = 1$. For the positive set, we only have set a bound on the lower proportion for the red color, specifically $l^P_{\text{red}} = \frac{3}{4}, u^P_{\text{red}} = 1$ and $l^P_{\text{blue}} = 0, u^P_{\text{blue}} = 1$. As the reduction of theorem (2.4.4) 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 $\frac{3q}{4q} = \frac{3}{4} \geq l^P_{\text{red}}$. 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 l^P_{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. □

We then have the following lemma:

Lemma 2.4.6. *Even if we do not specify the number of centers of each label (ignoring constraints*

(2.14d)), LCUL *is NP-hard.*

Proof. Similar to theorems (2.4.4,2.4.5) we follow the same reduction from X3C. This time we ignore constraint (2.14d) 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 $l^P_{\text{red}} = u^P_{\text{red}} = \frac{3}{4}$, $l^P_{\text{blue}} = u^P_{\text{blue}} = \frac{1}{4}$ and for the negative set we have $l^N_{\text{red}} = l^N_{\text{blue}} = 0$. $u^N_{\text{red}} = u^N_{\text{blue}} = 1$.

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 $(LB)_P = (UB)_P = 4q$, $l^P_{\text{red}} = u^P_{\text{red}} = \frac{3}{4}$, and $l^P_{\text{blue}} = u^P_{\text{blue}} = \frac{1}{4}$, 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. □

The proof of the theorem now follows immediately from the above lemmas (2.4.4,2.4.5,2.4.6) above. □

Having established the hardness of LCUL for different sets of constraints, we show that it is fixed-parameter tractable[11] for a constant number of labels. This immediately follows since a given 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.

[11] 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 [68] for more details.

Theorem 2.4.7. *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 (2.14b) or the constraint on the number of labels (2.14c), then the problem is solvable in polynomial time. However, if we only had constraint (2.14d) for the number of centers a label has, the problem is still NP-hard.

Theorem 2.4.8. *Even if number of labels* $m = \Omega(n)$*, the* LCUL *problem is solvable in polynomial time under constraint (2.14b) alone or constraint (2.14d) alone. However, it is NP-hard under constraint (2.14c) alone.*

Proof. Let us consider the color proportionality constraint (2.14b) 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 (2.14d). Again we assign each point to its closest center for an optimal cost. To satisfy constraints (2.14d), assuming the constraint parameters of (2.14d) 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 (2.14c). We simply follow the same reduction from theorems (2.4.4,2.4.5,2.4.6), 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. □

2.4.3.2 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 $\epsilon \geq 0$):

$$\min_{\phi,\ell} \Big(\sum_{j\in\mathcal{C}} d^p(j,\phi(j)) \Big)^{1/p} \tag{2.16a}$$

$$\forall L\in\mathcal{L}, \forall h\in\mathcal{H} : (r_h - \epsilon^A_{h,L}) \sum_{\substack{i\in S:\\ \ell(i)=L}} |\,\mathcal{C}_i\,| \leq \sum_{\substack{i\in S:\\ \ell(i)=L}} |\,\mathcal{C}^h_i\,| \leq (r_h + \epsilon'^A_{h,L}) \sum_{\substack{i\in S:\\ \ell(i)=L}} |\,\mathcal{C}_i\,| \tag{2.16b}$$

$$\forall L \in \mathcal{L} : (\alpha_L - \epsilon^B_L)n \leq \sum_{i\in S:\ell(i)=L} |\,\mathcal{C}_i\,| \leq (\alpha_L + \epsilon'^B_L)n \tag{2.16c}$$

$$\forall L \in \mathcal{L} : (\alpha_L - \epsilon^C_L)k \leq |S^L| \leq (\alpha_L + \epsilon'^C_L)k \tag{2.16d}$$

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 2.4.9. *The* CLP *problem is NP-hard even for the two color and two label case.*

Proof. We again follow a reduction for X3C. 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: Suppe 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 (7)) which always gives an optimal cost to the clustering and satisfies all constraints in expectation and further satisfies constraint (2.16d) 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 $p_L^i = \alpha_L$. Then in step 3, we apply dependent rounding, due to [69], to the probabilistic assignments to find the deterministic assignments. This leads to the following theorem:

Theorem 2.4.10. *Algorithm 7 gives an optimal clustering and satisfies constraints (2.16b,2.16c,2.16d) in expectation with (2.16d) 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 $p_L^i = \alpha_L$ for each center i. Now we prove that constraints (2.14b,2.14c,2.14d) hold in expectation over the assignments P_L^i. Note that P_L^i 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:

$$\mathbb{E}[\sum_{i \in S: \ell(i)=L} |\,\mathcal{C}_i\,|] = \mathbb{E}[\sum_{i \in S} |\,\mathcal{C}_i\,|\,P_L^i] = \sum_{i \in S} |\,\mathcal{C}_i\,|\,\mathbb{E}[P_L^i]$$

$$= \sum_{i \in S} |\,\mathcal{C}_i\,|\,p_L^i = \alpha_L \sum_{i \in S} |\,\mathcal{C}_i\,| = \alpha_L n$$

Clearly, constraint (2.16c) is satisfied. Through a similar argument we can show that the rest of the constraints also hold in expectation.

We have that $\forall L \in \mathcal{L} : |S^L| = \sum_{i \in S} P_L^i = \sum_{i \in S} \alpha_L = \alpha_L k$. 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 (2.16d) is satisfied in every run of the algorithm at a violation of at most 1. □

Algorithm 7 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 $p_L^i = \alpha_L$.
3: Apply dependent rounding [69] to probabilistic assignments p_L^i to get the deterministic assignments P_L^i

We note that dependent rounding enjoys the **Marginal Probability** property which means that $\Pr[P_L^i = 1] = p_L^i$. 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 $\forall L \in \mathcal{L} : \sum_{i \in S} P_L^i \in \{\lfloor \sum_{i \in S} p_L^i \rfloor, \lceil \sum_{i \in S} p_L^i \rceil\}$ which leads us to satisfy constraint (2.16d) 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 2.4.4.2) that dependent rounding is much better concentrated around the expected value, especially for constraint (2.16c) for the number of points in each label.

2.4.4 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 [70]. We evaluate the performance of our algorithms over a collection of datasets from the UCI repository [71]. 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 $h \in \mathcal{H}$ with population proportion $r_h = \frac{|\mathcal{C}^h|}{|\mathcal{C}|}$ we set the the upper and lower proportion bounds to $l_h = (1-\delta)r_h$ and $u_h = (1+\delta)r_h$, 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 $\mathrm{PoF} = \frac{\text{fair solution cost}}{\text{color-blind solution cost}}$ where fair solution cost is the cost of the fair variant the 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 (2.14b), we consider a label L and define $\Delta_h^L \in [0,1]$ where Δ_h^L 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: $(l_h^L - \Delta_h^L)\sum_{i\in S:\ell(i)=L}|\mathcal{C}_i| \le \sum_{i\in S:\ell(i)=L}|\mathcal{C}_i^h| \le (u_h^L + \Delta_h^L)\sum_{i\in S:\ell(i)=L}|\mathcal{C}_i|$, having

found Δ_h^L we report Δ_{color} where $\Delta_{\text{color}} = \max_{\{h\in\mathcal{H}, l\in\mathcal{L}\}} \Delta_h^L$. 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 [44] 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 [40] 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 [21]. 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 **CreditCard** 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 **CreditCard** 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 **CreditCard** dataset we use age and 12 other

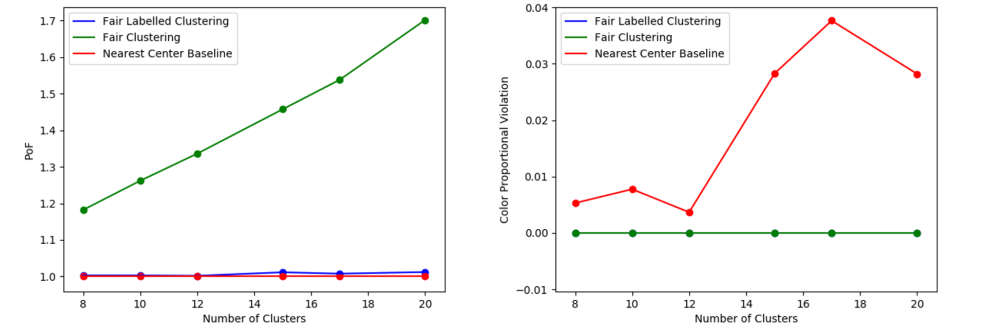
financial entries as coordinates.

2.4.4.1 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 $\geq 1{,}100$ 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 2.13 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 2.5.6. 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 **CreditCard** datasets. We can clearly see that

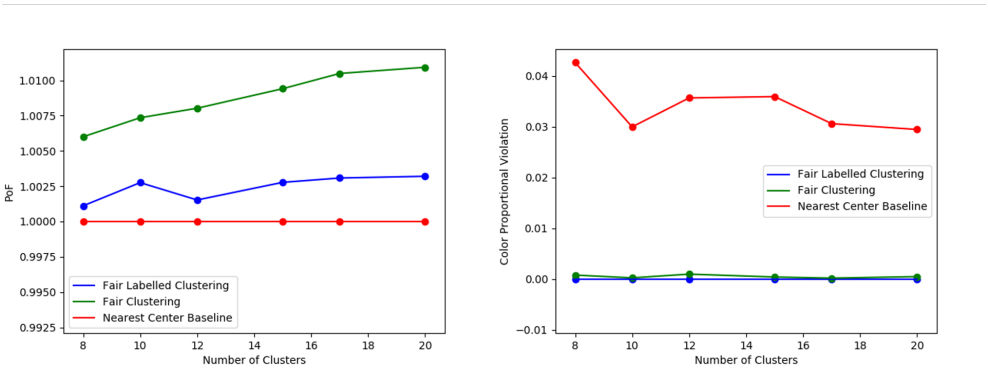
although the (fairness-agnostic) solution has the smallest cost it has large color violation. We also see that our algorithm unlike fair clustering achieves fairness but at a much lower PoF.



(a): PoF (b): Color Proportional Violation

Figure 2.13: **Adult** dataset results (a):PoF, (b):Δ_{color}

CreditCard 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 $\geq$ 300,000 we assign the center a positive label and a negative one otherwise. Figure 2.14 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 2.14: **CreditCard** dataset results (a):PoF, (b):Δ_{color}

As mentioned in section 2.4.2.2, algorithm (6) 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 2.15 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.

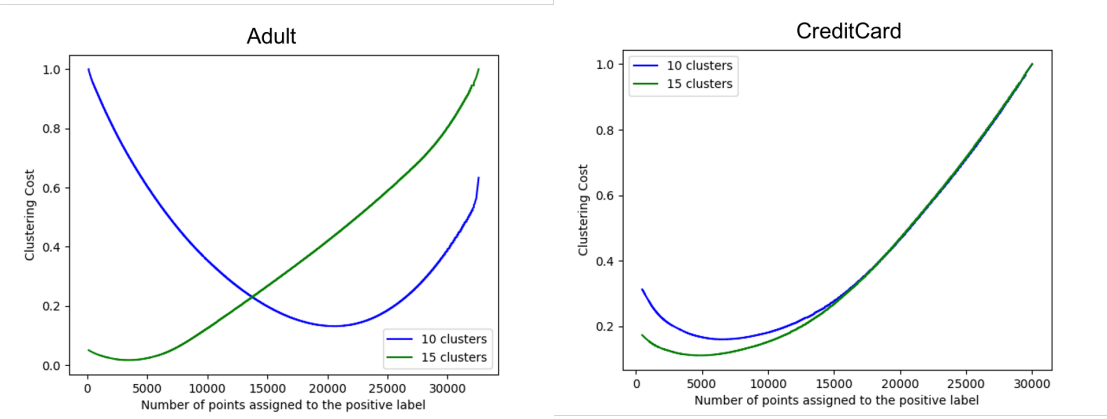


Figure 2.15: A plot of $|\phi^{-1}(P)|$ vs the clustering cost (normalized by the maximum cost obtained).

2.4.4.2 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 7) to two baselines: (1) Nearest Center with Random Assignment (**NCRA**) and (2) Fair Clustering (**FC**). We refer to our algorithm (block 7) 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 $\Delta_{\text{center/label}}$.

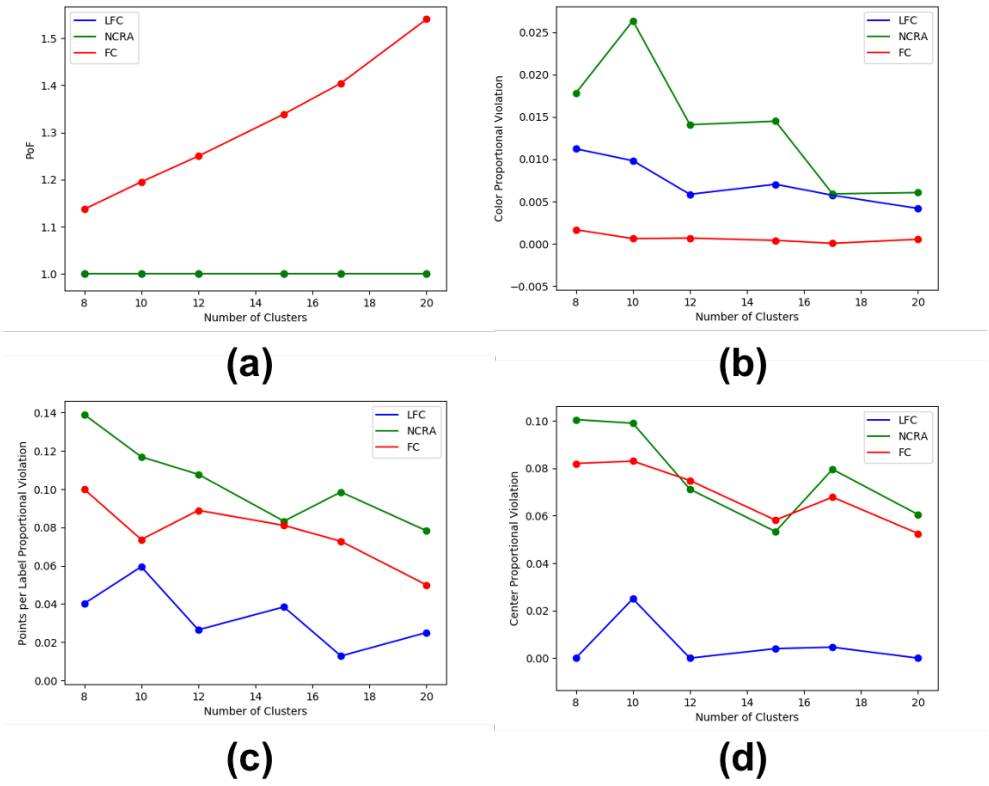


Figure 2.16: LCUL results on the **Adult** dataset. (a):PoF, (b):Δ_{color}, (c):$\Delta_{\text{points/label}}$,(d):$\Delta_{\text{center/label}}$.

Figures 2.16 and 2.17 show the results for **Adult** and **CreditCard**. 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

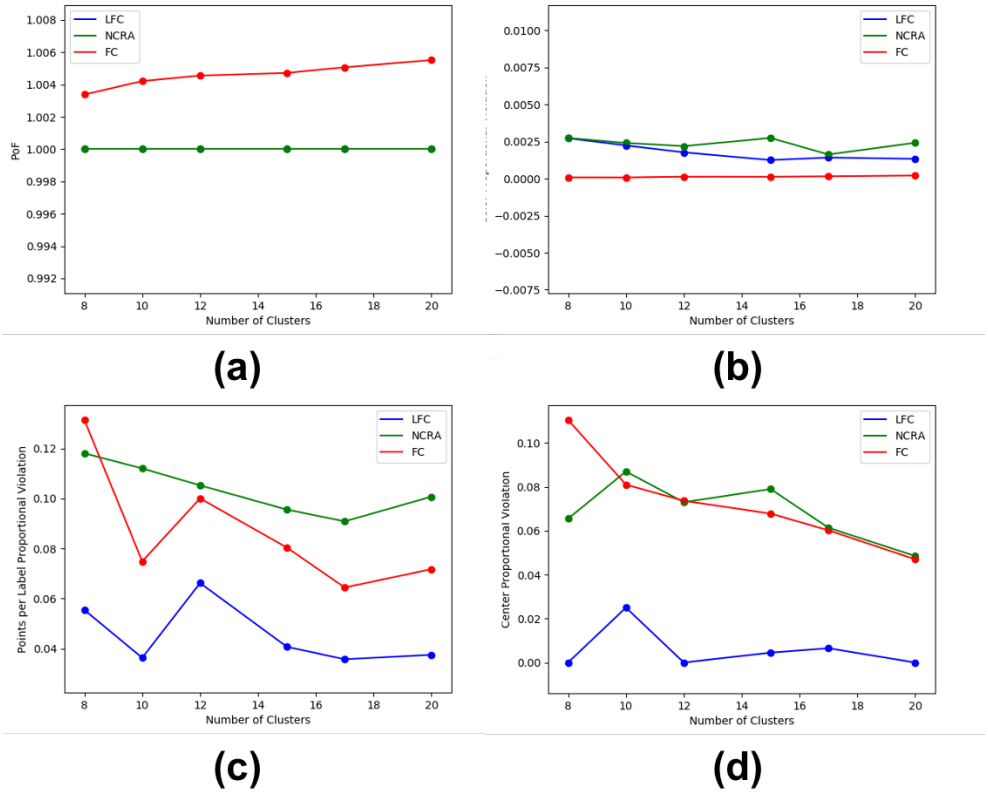


Figure 2.17: LCUL results on the **CreditCard** dataset. (a):PoF, (b):Δ_{color}, (c):$\Delta_{\text{points/label}}$,(d):$\Delta_{\text{center/label}}$.

has almost no violation whereas the **NCRA** and labeled clustering have small but noticeable violations. For the 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.

2.4.4.3 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 the centers with the k-means algorithm[12], 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 2.18) and that even for 500,000 points it takes less than 90 seconds. We note in contrast that the fair clustering algorithm of [40] 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 [72, 73]. The fact that our algorithm performs relatively well run-time wise is worthy of noting.

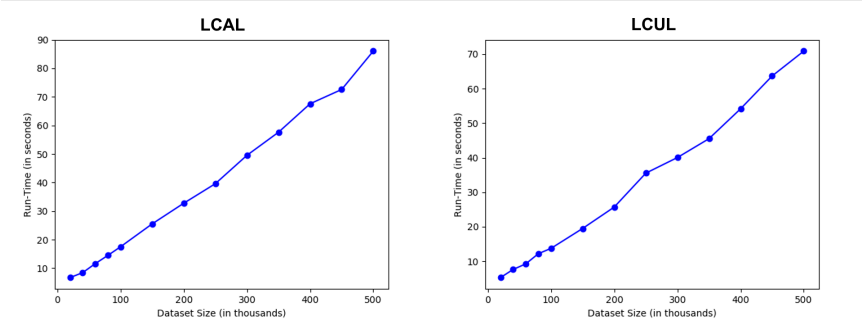


Figure 2.18: Dataset size vs algorithm Run-Time: (left) LCAL, (right) LCUL.

2.5 Doubly Constrained Fair Clustering

There has been a significant number of fairness notions that have been introduced in fair clustering [20], it is possible to list at least seven different notions. Although each notion is well-

[12]We choose $k = 5$ for all different dataset sizes.

justified, it is always motivated in a disjoint manner where the other fairness notions are ignored. Ideally, one would desire a single clustering of the data which satisfies a collection of fairness notions instead of having different clusterings for different fairness notions. A similar question was investigated in fair classification [74, 75] where it was shown that unless the given classification instance satisfies restrictive conditions, the two desired fairness objectives of calibration and balance cannot be simultaneously satisfied. One would expect that such a guarantee would also hold in fair clustering. For various constraints it can be shown that they are in fact at odds with one another. However, it is also worthwhile on the other hand to ask *if some fair clustering constraints are more compatible with one another, and how one can satisfy both simultaneously?*

We take a first step towards understanding this question. In particular, we consider two specific group fairness constraints (1) **GF**: The group fair clustering (**GF**) of [21] which roughly states that clusters should have close to population level proportions of each group (this is the constraint the previous sections were essentially concerned with) and (2) **DS**: The diversity in center selection (**DS**) constraint [22] which roughly states that the selected centers in the clustering should similarly include close to population level proportions of each group. We note that although these two definitions are both concerned with group memberships, the fact that they apply at different "levels" (clustered points vs selected centers) makes the algorithms and guarantees that are applicable for one problem not applicable to the other, certainly not in an obvious way. Further, both of these notions are motivated by disparate impact [9] which essentially states that different groups should receive the same treatment. Therefore, it is natural to consider the intersection of both definitions (**GF**+**DS**). We show that by post-processing any solution satisfying one constraint then we can always satisfy the intersection of both constraints. At a more precise level, we show that an α-approximation algorithm for one constraint results in an approximation

algorithm for the intersection of the constraints with only a constant degradation to approximation ratio α. Additionally, we study the degradation in the clustering cost and show that imposing **DS** on a **GF** solution leads to a bounded degradation of the clustering cost while the reverse is not true. Moreover, we show that both **GF** and **DS** are incompatible (having an empty feasible set) with a set of distance-based fairness constraints that were introduced in the literature. Finally, we validate our findings experimentally.

2.5.1 Preliminary Remarks, Definitions and Symbols

Here we are only concerned with the k-center clustering which minimizes the maximum distance between a point and its assigned center. Formally, we have:

$$\min_{S:|S|\leq k,\phi} \max_{j\in\mathcal{C}} d(j,\phi(j)) \tag{2.17}$$

We now formally revise the group fair clustering (**GF**) and introduce the diverse center selection (**DS**) problems:

Group Fair Clustering [1, 14, 21, 39, 40]: Minimize objective (2.17) subject to proportional demographic representation in each cluster. Specifically, $\forall i \in S, \forall h \in \mathcal{H} : \beta_h |\, C_i \,| \leq |\, C_i^h \,| \leq \alpha_h |\, C_i \,|$ where β_h and α_h are pre-set upper and lower bounds for the demographic representation of color h in a given cluster. Further, C_i is the i^{th} cluster.

Diverse Center Selection [22, 23, 76]: Minimize objective (2.17) subject to the set of centers S satisfying demographic representation. Specifically, denoting the number of centers from demographic (color) h by $k_h = |S \cap \mathcal{C}^h|$, then as done in [23] it must satisfy $k_h^l \leq k_h \leq k_h^u$ where

k_h^l and k_h^u are lower and upper bounds set for the number of centers of color h, respectively.

Importantly, throughout we have $\forall h \in \mathcal{H} : \beta_h > 0$. Further, for **GF** we consider solutions that could have violations to the constraints as done in the previous sections. Specifically, a given a solution (S, ϕ) has an additive violation of ρ **GF** if ρ is the smallest number such that the following holds: $\forall i \in S, \forall h \in \mathcal{H} : \beta_h |C_i| - \rho \leq |C_i^h| \leq \alpha_h |C_i| + \rho$. We denote the problem of minimizing the k-center objective while satisfying both the **GF** and **DS** constraints as **GF+DS**.

Why Consider GF and DS in Particular? There are two reasons to consider the **GF** and **DS** constraints in particular. First, from the point of view of the application both **GF** and **DS** are concerned with demographic (group) fairness. Further, they are both specifically focused on the representation of groups, i.e. the proportions of the groups (colors) in the clusters for **GF** and in the selected center for **DS**. Second, they are both "distance-agnostic", i.e. given a clustering solution one can decide if it satisfies the **GF** or **DS** constraints without having access to the distance between the points.

2.5.2 Algorithms for **GF+DS**

2.5.2.1 Active Centers

We start by observing the fact that if we wanted to satisfy both **GF** and **DS** simultaneously, then we should make sure that all centers are *active* (having non-empty clusters). More precisely, given a solution (S, ϕ) then the **DS** constraints should be satisfied further $\forall i \in S : |C_i| > 0$, i.e. every center in S should have some point assigned to it and therefore not forming an empty cluster. The following example clarifies this:

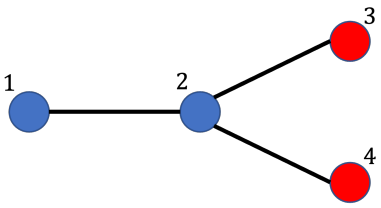


Figure 2.19: In this graph the distance between the points is the path distance.

Example: Consider Figure 2.19. Suppose we have $k = 2$ and we wish to satisfy the **GF** and **DS** constraints with equal red to blue representation. **DS** requires one blue and one red center. Further, each cluster should have $|C_i^{\text{blue}}| = |C_i^{\text{red}}| = \frac{1}{2}|C_i|$ to satisfy **GF**. Consider the following solution $S_1 = \{2, 4\}$ and ϕ_1 which assigns all points to point 2 including point 4. This satisfies **GF** and **DS**. Since we have one blue center and one red center. Further, the cluster of center 4 has no points and therefore $0 = |C_i^{\text{blue}}| = |C_i^{\text{red}}| = \frac{1}{2}|C_i|$. Another solution would have $S_2 = S_1 = \{2, 4\}$ but with ϕ_2 assigning points 2 and 3 to center 2 and points 1 and 4 to center 4. This would also clearly satisfy the **GF** and **DS** constraints.

There is a clear issue in the first solution which is that although center 4 is included in the selection it has no points assigned to it (it is an empty cluster). This makes it functionally non-existent. This is why the definition should only count active centers.

This issue of active centers did not appear before in **DS** [22, 23], the reason behind this is that it is trivial to satisfy when considering only the **DS** constraint since each center is assigned all the points closest to it. This implies that the center will at least be assigned to itself, therefore all centers in a **DS** solution are *active*. However, we cannot simply assign each point to its closest center when the **GF** constraints are imposed additionally as the colors of the points have to satisfy the upper and lower proportion bounds of **GF**.

2.5.2.2 The DIVIDE Subroutine

Here we introduce the DIVIDE subroutine (block 8) which is used in constructing algorithms for converting solutions that only satisfy **DS** or **GF** into solutions that satisfy **GF+DS**. DIVIDE takes a set of points C (which is supposed to be a single cluster) with center i along with a subset of chosen points Q ($Q \subset C$). The entire set of points is then divided among the points Q forming $|Q|$ many new non-empty (active) clusters. Importantly, the points of each color are divided among the new centers in Q so that the additive violation increases by at most 2. See Figure 2.20 for an intuitive illustration.

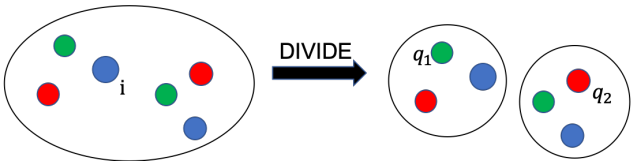


Figure 2.20: Illustration of DIVIDE subroutine.

Here we use the symbol q to index a point in the set Q. Importantly, the numbering starts with 0 and ends with $|Q| - 1$.

Before we give the guarantees for DIVIDE, we note the following lemma:

Lemma 2.5.1. *Given a fractional solution* $\mathbf{x}^{frac}$ *that satisfies the* ***GF*** *constraints at an additive violation of at most* ρ*, then if there exists an integral solution* $\mathbf{x}^{integ}$ *that satisfies:*

$$\forall q \in Q : \left\lfloor \sum_{j \in C} x_{qj}^{frac} \right\rfloor \leq \sum_{j \in C} x_{qj}^{integ} \leq \left\lceil \sum_{j \in C} x_{qj}^{frac} \right\rceil \tag{2.18}$$

$$\forall q \in Q, h \in \mathcal{H} : \left\lfloor \sum_{j \in C^h} x_{qj}^{frac} \right\rfloor \leq \sum_{j \in C^h} x_{qj}^{integ} \leq \left\lceil \sum_{j \in C^h} x_{qj}^{frac} \right\rceil \tag{2.19}$$

Then this integral solution $\mathbf{x}^{integ}$ *satisifies the* ***GF*** *constraints at an additive violation of at most*

Algorithm 8 DIVIDE

Input: Set of points C with center $i \in C$, Subset of points Q ($Q \subset C$) of cardinality $|Q|$.
Output: An assignment function $\phi : C \to Q$.

if $|Q| = 1$ **then**
  Assign all points C to the single center in Q.
else
  Set firstIndex $= 0$.
  for $h \in \mathcal{H}$ **do**
    Set: $T_h = \frac{|C^h|}{|Q|},\ b_h = T_h - |Q|\,\lfloor T_h \rfloor$, count $= 0$
    Set: $q =$ firstIndex
    while count $\leq |Q| - 1$ **do**
      if $b_h > 0$ **then**
        Assign $\lceil T_h \rceil$ many points of color h in C to center q.
        Update $b_h = b_h - 1$.
        Update firstIndex $= (\text{firstIndex} + 1) \mod |Q|$.
      else
        Assign $\lfloor T_h \rfloor$ many points of color h in C to center q.
      end if
      Update $q = (q + 1) \mod |Q|$, count $=$ count $+ 1$.
    end while
  end for
end if

$\rho + 2$.

Proof. Since the fractional solution satisfies the **GF** constraints at an additive violation of ρ, then we have the following:

$$-\rho + \left(\beta_h \sum_{j \in C} x_{qj}^{\text{frac}}\right) \leq \sum_{j \in C^h} x_{qj}^{\text{frac}} \leq \left(\alpha_h \sum_{j \in C} x_{qj}^{\text{frac}}\right) + \rho$$

We start with the upper bound:

$$\begin{aligned}
\sum_{j \in C^h} x_{qj}^{\text{integ}} &\leq \left\lceil \sum_{j \in C^h} x_{qj}^{\text{frac}} \right\rceil \\
&\leq \sum_{j \in C^h} x_{qj}^{\text{frac}} + 1 \\
&\leq \alpha_h \sum_{j \in C} x_{qj}^{\text{frac}} + \rho + 1 \\
&\leq \alpha_h (\sum_{j \in C} x_{qj}^{\text{integ}} + 1) + \rho + 1 \\
&\leq \alpha_h \sum_{j \in C} x_{qj}^{\text{integ}} + (\alpha_h + \rho + 1) \\
&\leq \alpha_h \sum_{j \in C} x_{qj}^{\text{integ}} + (\rho + 2)
\end{aligned}$$

Now we do the lower bound:

$$\begin{aligned}
\sum_{j \in C^h} x_{qj}^{\text{integ}} &\geq \left\lfloor \sum_{j \in C^h} x_{qj}^{\text{frac}} \right\rfloor \\
&\geq \sum_{j \in C^h} x_{qj}^{\text{frac}} - 1 \\
&\geq \beta_h \sum_{j \in C} x_{qj}^{\text{frac}} - \rho - 1 \\
&\geq \beta_h (\sum_{j \in C} x_{qj}^{\text{integ}} - 1) - (\rho + 1) \\
&\geq \beta_h \sum_{j \in C} x_{qj}^{\text{integ}} - (\beta_h + \rho + 1) \\
&\geq \beta_h \sum_{j \in C} x_{qj}^{\text{integ}} - (\rho + 2)
\end{aligned}$$

□

Now we prove the following about DIVIDE:

Lemma 2.5.2. *Given a non-empty cluster C with center i and radius R that satisfies the **GF** constraints at an additive violation of ρ and a subset of points Q ($Q \subset C$). Then the clustering (Q, ϕ) where $\phi =$ DIVIDE(C, Q) has the following properties: (1) The **GF** constraints are satisfied at an additive violation of at most $\frac{\rho}{|Q|} + 2$. (2) Every center in Q is* active. *(3) The clustering cost is at most $2R$. If $|Q| = 1$ then guarantee (1) is for the additive violation is at most ρ.*

Proof. We first consider the case where $|Q| > 1$. We prove the following claim:

Claim 2.5.3. *For the fractional assignment $\{x_{qj}^{frac}\}_{q \in Q, j \in C}$ such that:*

$$\forall q \in Q, \forall h \in \mathcal{H} : \sum_{j \in C^h} x_{qj}^{frac} = \frac{|C^h|}{|Q|} = T_h$$

*It holds that: (1) $\forall q \in Q : \sum_{j \in C} x_{qj}^{frac} \geq 1$, (2) **GF** constraints are satisfied at an additive violation of $\frac{\rho}{|Q|}$.*

Proof. Now we prove the first property

$$\forall q \in Q : \sum_{j \in C} x_{qj}^{\text{frac}} = \sum_{h \in \mathcal{H}} \sum_{j \in C^h} x_{qj}^{\text{frac}} = \frac{1}{|Q|} \sum_{h \in \mathcal{H}} |C^h| = \frac{|C|}{|Q|} \geq 1 \quad (\text{since } Q \subset C) \tag{2.20}$$

.

Since the **GF** constraints given center i are satisfied at an additive violation of ρ, then we have:

$$\forall h \in \mathcal{H} : -\rho + \beta_h |C| \leq |C^h| \leq \alpha_h |C| + \rho \tag{2.21}$$

Therefore, since the amount of color for each center in Q with the fractional assignment can be obtained by dividing by $|Q|$, then we have:

$$\forall h \in \mathcal{H}, \forall q \in Q : -\frac{\rho}{|Q|} + \beta_h \sum_{j \in C} x_{qj}^{\text{frac}} \leq \sum_{j \in C^h} x_{qj}^{\text{frac}} \leq \alpha_h \sum_{j \in C} x_{qj}^{\text{frac}} + \frac{\rho}{|Q|} \tag{2.22}$$

Therefore the **GF** constraints are satisfied at an additive violation of $\frac{\rho}{|Q|}$. □

Denoting the assignment ϕ resulting from DIVIDE by $\{x_{qj}^{\text{integ}}\}_{q \in Q, j \in C}$, then the following claim holds:

Claim 2.5.4.

$$\forall q \in Q : \left\lfloor \sum_{j \in C} x_{qj}^{frac} \right\rfloor \leq \sum_{j \in C} x_{qj}^{integ} \leq \left\lceil \sum_{j \in C} x_{qj}^{frac} \right\rceil$$

$$\forall q \in Q, h \in \mathcal{H} : \left\lfloor \sum_{j \in C^h} x_{qj}^{frac} \right\rfloor \leq \sum_{j \in C^h} x_{qj}^{integ} \leq \left\lceil \sum_{j \in C^h} x_{qj}^{frac} \right\rceil$$

Proof. For any color h we have $|C_h| = a_h|Q| + b_h$ where a_h and b_h are non-negative integers and b_h is the remainder of dividing $|C_h|$ by Q ($b_h \in \{0, 1, \ldots, |Q|-1\}$). It follows that $\sum_{j \in C^h} x_{qj}^{\text{frac}} = T_h = a_h + \frac{b_h}{|Q|}$. DIVIDE gives each center either $\sum_{j \in C^h} x_{qj}^{\text{integ}} = a_h = \lfloor T_h \rfloor = \left\lfloor \sum_{j \in C^h} x_{qj}^{\text{frac}} \right\rfloor$ or $\sum_{j \in C^h} x_{qj}^{\text{integ}} = a_h + 1 = \lceil T_h \rceil = \left\lceil \sum_{j \in C^h} x_{qj}^{\text{frac}} \right\rceil$. This proves the second condition.

For the first condition, note that $|C| = \sum_{h \in \mathcal{H}} (a_h|Q| + b_h) = (\sum_{h \in \mathcal{H}} a_h)|Q| + a|Q| + b$ where we set $\sum_{h \in \mathcal{H}} b_h = a|Q| + b$ with a and b being non-negative integers. b is the remainder and has values in $\{0, 1, \ldots, |Q| - 1\}$. Accordingly, the sum of the remainders across the colors is $a|Q| + b$. Since the remainders are added "successivly" across the centers (see Figure 2.20) and a is divisible by $|Q|$, then for any center $q \in Q$ either $\sum_{j \in C} x_{qj}^{\text{integ}} = (\sum_{h \in \mathcal{H}} a_h) + a$ or $\sum_{j \in C} x_{qj}^{\text{integ}} = (\sum_{h \in \mathcal{H}} a_h) + a + 1$. Note that $\sum_{j \in C} x_{qj}^{\text{frac}} = \sum_{h \in \mathcal{H}} T_h = (\sum_{h \in \mathcal{H}} a_h) + a + \frac{b}{|Q|}$.

Therefore, $\left\lfloor \sum_{j \in C} x_{qj}^{\text{frac}} \right\rfloor = (\sum_{h \in \mathcal{H}} a_h) + a$ and $\left\lceil \sum_{j \in C} x_{qj}^{\text{frac}} \right\rceil = (\sum_{h \in \mathcal{H}} a_h) + a + 1$. This proves, the first condition. □

By Claim 2.5.4 and Lemma 2.5.1 it follows that for each center $q \in Q$ the assignment $\{x_{qj}^{\text{integ}}\}_{q \in Q, j \in C}$ satisfies the **GF** constraints at an additive violation of $\frac{\rho}{|Q|} + 2$, this proves the first guarantee.

By Claim 2.5.4 and guarentee (1) of Claim 2.5.3, then $\forall q \in Q : \sum_{j \in C} x_{qj}^{\text{integ}} \geq \left\lfloor \sum_{j \in C} x_{qj}^{\text{frac}} \right\rfloor \geq 1$. Therfeore, every center $q \in Q$ is *active* proving the second guarantee.

Guarantee (3) follows since $\forall j \in C : d(j, \phi(j)) \leq d(j, i) + d(i, \phi(j)) \leq 2R$.

Now if $|Q| = 1$, then guarantee (2) follows since the cluster C is non-empty. Guarantee (3) follows similarly to the above. The additive violation in the **GF** constraint on the other hand is ρ since the single center Q has the exact set of points that were assigned to the original center i. □

2.5.2.3 Solving **GF+DS** using a **DS** Algorithm

Here we show an algorithm that gives a bounded approximation for **GF+DS** using an approximation algorithm for **DS**. Algorithm 9 works by first calling an $\alpha_{\textbf{DS}}$-approximation algorithm resulting in a solution $(\bar{S}, \bar{\phi})$ that satisfies the **DS** constraints, then it solves an assignment problem using the ASSIGNMENTGF algorithm (shown in 10) where points are routed to the centers $\bar{S}$ to satisfy the **GF** constraint. The issue is that some of the centers in $\bar{S}$ may become closed and as a result the solution may no longer satisfy the **DS** constraints. Therefore, we have a final step where more centers are opened using the DIVIDE subroutine to satisfy the **DS** constraints while still satisfying the **GF** constraints at an additive violation and having a bounded increase to

the clustering cost.

Algorithm 9 DSTO**GF+DS**

Input: Points $\mathcal{C}$, Solution $(\bar{S}, \bar{\phi})$ with clusters $\{C_i, \ldots, C_{\bar{k}}\}$ satisfying the **DS** constraints with $|\bar{S}| = \bar{k} \leq k$ of approximation ratio $\alpha_{\textbf{DS}}$ for the **DS** clustering problem.
Output: Solution (S, ϕ) satisfying the **GF** and **DS** constraints simultaneously.

(S', ϕ') =ASSIGNMENTGF$(\bar{S}, \mathcal{C})$
Update the set of centers S' by deleting all non-active centers (which have no points assigned to them). Let $\{C'_1, \ldots, C'_{k'}\}$ be the (non-empty) clusters of the solution (S', ϕ') with $|S'| = k' \leq \bar{k}$.
Set $\forall h \in \mathcal{H} : s_h = |S' \cap \mathcal{C}^h|$, Set $\forall i \in S : Q_i = \{i\}$
while $\exists h \in \mathcal{H}$ such that $s_h < k^l_h$ **do**
    Pick a color h_0 such that $s_{h_0} < k^l_{h_0}$.
    Pick a center $i \in S'$ where there exists a point of color h_0.
    Pick a point j_{h_0} of color h_0 in cluster C'_i
    Set $Q_i = Q_i \cup \{j_{h_0}\}$.
    Update $s_{h_0} = s_{h_0} + 1$.
end while
for $i \in S'$ **do**
    $\phi_i =$ DIVIDE(C'_i, Q_i).
    $\forall j \in C'_i$: Set $\phi(j) = \phi_i(j)$.
end for
Set $S = S' \cup \big(\cup_{i \in S'} Q_i \big)$.

Algorithm 10 ASSIGNMENTGF

Input: Set of centers S, Set of Points C.
Output: An assignment function $\phi : C \to S$.

Using binary search over the distance matrix, find the smallest radius R such that $LP(C, S, R)$ in (2.23) is feasible and call the solution $\mathbf{x}^*$.
Solve MAXFLOWGF$(\mathbf{x}^*, C, S)$ and call the solution $\bar{\mathbf{x}}^*$.

ASSIGNMENTGF works by solving a linear program (2.23) to find a clustering which ensures that (1) each cluster has at least a β_h fraction and at most an α_h fraction of its points belonging to color h, and (2) the clustering assigns each point to a center that is within a minimum possible distance R. While the resulting LP solution could be fractional, the last step of ASSIGNMENTGF uses MAXFLOWGF which is an algorithm for rounding an LP solution to

valid integral assignments at a bounded degradation to the **GF** guarantees and no increase to the clustering cost. See [1,39] for details on MAXFLOWGF and its guarantees.

LP(C, S, R) :

$$\forall j \in C, \forall i \in S : x_{ij} = 0 \quad \text{if } d(i,j) > R \tag{2.23a}$$

$$\forall h \in \mathcal{H}, \forall i \in S : \beta_h \sum_{j \in C} x_{ij} \leq \sum_{j \in C^h} x_{ij} \leq \alpha_h \sum_{j \in C} x_{ij} \tag{2.23b}$$

$$\forall j \in C : \sum_{i \in S} x_{ij} = 1 \tag{2.23c}$$

$$\forall j \in C, \forall i \in S : x_{ij} \in [0,1] \tag{2.23d}$$

To establish the guarentees we start with the following lemma:

Lemma 2.5.5. *Solution (S', ϕ') of line (3) in algorithm 9 has the following properties: (1) It satisfies the **GF** constraint at an additive violation of 2, (2) It has a clustering cost of at most $(1 + \alpha_{\textbf{DS}})R^*_{\textbf{GF+DS}}$ where $R^*_{\textbf{GF+DS}}$ is the optimal clustering cost (radius) of the optimal solution for **GF+DS**, (3) The set of centers S' is a subset (possibly proper subset) of the set of centers $\bar{S}$, i.e. $S' \subset S$.*

Proof. We begin with the following claim which shows that there exists a solution that only uses centers from $\bar{S}$ to satisfy the **GF** constraints exactly and at a radius of at most $(1 + \alpha_{\textbf{DS}})R^*_{\textbf{GF+DS}}$. Note that this claim has non-constructive proof, i.e. it only proves the existence of such a solution:

Claim 2.5.6. *Given the set of centers $\bar{S}$ resulting from the $\alpha_{\textbf{DS}}$-approximation algorithm, then there exists an assignment ϕ_0 from points in $\mathcal{C}$ to centers in $\bar{S}$ such that the following holds: (1) The **GF** constraint is exactly satisfied (additive violation of 0). (2) The clustering cost is at most*

$(1+\alpha_{\textbf{\textit{DS}}})R^*_{\textbf{\textit{GF+DS}}}$.

Proof. Let $(S^*_{\textbf{GF+DS}}, \phi^*{}_{\textbf{GF+DS}})$ be an optimal solution to the **GF+DS** problem. $\forall i \in S^*_{\textbf{GF+DS}}$ let $N(i) = \arg\min_{\bar{i}\in\bar{S}} d(i,\bar{i})$, i.e. $N(i)$ is the nearest center in $\bar{S}$ to center i (ties are broken using the smallest index). ϕ_0 is formed by assigning all points which belong to center $i \in S^*_{\textbf{GF+DS}}$ to $N(i)$. More formally, $\forall j \in \mathcal{C} : \phi^*{}_{\textbf{GF+DS}}(j) = i$ we set $\phi_0(j) = N(i)$. Note that it is possible for more than one center i in $S^*_{\textbf{GF+DS}}$ to have the same nearest center in $\bar{S}$. We will now show that ϕ_0 satisfies the **GF** constraint exactly. Note first that if a center $\bar{i} \in \bar{S}$ has not been assigned any points by ϕ_0, then it is empty and trivially satisfies the **GF** constraint exactly. Therefore, we assume that $\bar{i}$ has a non-empty cluster. Denote by $N^{-1}(\bar{i})$ the set of centers $i \in S^*_{\textbf{GF+DS}}$ for which $\bar{i}$ is the nearest center, then by the fact that every cluster in $(S^*_{\textbf{GF+DS}}, \phi^*{}_{\textbf{GF+DS}})$ satisfies the **GF** constraint exactly we have:

$$\beta_h \leq \min_{i\in N^{-1}(\bar{i})} \frac{|\,C_i^h\,|}{|\,C_i\,|} \leq \frac{\sum_{i\in N^{-1}(\bar{i})}|\,C_i^h\,|}{\sum_{i\in N^{-1}(\bar{i})}|\,C_i\,|} = \frac{|C_{\bar{i}}^h|}{|C_{\bar{i}}|} \leq \max_{i\in N^{-1}(\bar{i})} \frac{|\,C_i^h\,|}{|\,C_i\,|} \leq \alpha_h \tag{2.24}$$

The proves guarantee (1) of the lemma. Now we prove guarantee (2), we denote by $R^*_{\textbf{DS}}$ the optimal clustering cost for the **DS** constrained problem. We can show that $\forall j \in \mathcal{C}$:

$$\begin{aligned}
d(j,\phi_0(j)) &\leq d(j,\phi^*{}_{\textbf{GF+DS}}(j)) + d(\phi^*{}_{\textbf{GF+DS}}(j),\phi_0(j)) \\
&\leq d(j,\phi^*{}_{\textbf{GF+DS}}(j)) + d(\phi^*{}_{\textbf{GF+DS}}(j),N(\phi^*{}_{\textbf{GF+DS}}(j))) \quad (\text{ since } \phi_0(j) = N(\phi^*{}_{\textbf{GF+DS}}(j)) \text{) } \\
&\leq R^*_{\textbf{GF+DS}} + \alpha_{\textbf{DS}} R^*_{\textbf{DS}} \qquad\qquad (\text{ since } \bar{S} \text{ is an } \alpha_{\textbf{DS}}\text{-approximation for } \textbf{DS} \text{) } \\
&\leq (1+\alpha_{\textbf{DS}}) R^*_{\textbf{GF+DS}}
\end{aligned}$$

Where the last holds since $R^*_{\textbf{DS}} \leq R^*_{\textbf{GF+DS}}$ because the set of solutions constrained by **DS** is a

subset of the set of solutions constrained by **GF+DS**. □

Now we can prove the lemma. By the above claim, it follows that when ASSIGNMENTGF is called, the LP solution from line (3) of algorithm block 10 satisfies: (1) The **GF** constraints exactly and (2) Has a clustering cost of at most $(1 + \alpha_{\textbf{DS}})R^*_{\textbf{GF+DS}}$. This is because LP (2.23) includes all integral assignments from $\mathcal{C}$ to $\bar{S}$ including ϕ_0. Since this LP assignment is fed to MAXFLOWGF it follows that the final solution satisfies: (1) The **GF** constraint at an additive violation of 2, (2) Has a clustering cost of at most $(1 + \alpha_{\textbf{DS}})R^*_{\textbf{GF+DS}}$. Guarantee (3) holds since some centers may become closed (assigned no points) and therefore $S' \subset \bar{S}$ (possibly being a proper subset). □

Theorem 2.5.7. *Given an $\alpha_{\textbf{DS}}$-approximation algorithm for the **DS** problem, then we can obtain an $2(1+\alpha_{\textbf{DS}})$-approximation algorithm that satisfies **GF** at an additive violation of 3 and satisfies **DS** simultaneously.*

Proof. By Lemma 2.5.5 above, the set of centers S' is a subset (possibly proper) subset of S and therefore the **DS** constraints may no longer be satisfied. Algorithm 9 select points from each color h so that when they are added to S', then for each color h the set of centers is at least $\beta_h k$. Since these new centers are opened using the DIVIDE subroutine then it follows that they are all active (guarantee (2) of Lemma 2.5.2).

Further, by guarantee (3) of Lemma 2.5.2 for DIVIDE we have for any point j assigned to a new center q that $d(j, q) \leq 2d(j, \phi'(j)) \leq 2(1 + \alpha_{\textbf{DS}})R^*_{\textbf{GF+DS}}$.

Finally, by guarantee (1) of Lemma 2.5.2 DIVIDE is called over a cluster that satisfies **GF** at an additive violation of 2 and therefore the resulting additive violation is at most $\max\{2, \frac{2}{|Q_i|}+2\}$. Since $2 \leq \frac{2}{|Q_i|} + 2 \leq \frac{2}{2} + 2 = 3$. The additive violation is at most 3. □

Remark: If in algorithm 9 no center is deleted in line (4) because it forms an empty cluster, then by Lemma 2.5.5 the approximation ratio is $1 + \alpha_{\mathbf{DS}}$ which is an improvement by a factor of 2. Further, the additive violation for **GF** is reduced from 3 to 2.

2.5.3 Solving **GF+DS** using a **GF** Solution

Algorithm 11 GFTO**GF+DS**

Input: Points $\mathcal{C}$, Solution $(\bar{S}, \bar{\phi})$ with clusters $\{\bar{C}_i, \ldots, \bar{C}_{\bar{k}}\}$ satisfying the **GF** constraints with $|\bar{S}| = \bar{k} \leq k$.
Output: Solution (S, ϕ) satisfying the **GF** and **DS** constraints simultaneously.

Initialize: $\forall h \in \mathcal{H} : s_h = 0, \forall i \in \bar{S} : Q_i = \{\}$.
for $i \in \bar{S}$ **do**
  if $\exists h \in \mathcal{H} : s_h < k_h^l$ **then**
    Let h_0 be a color such that $s_{h_0} < k_{h_0}^l$
  else
    Pick h_0 such that $s_{h_0} + 1 \leq k_{h_0}^u$.
  end if
  Pick a point j_{h_0} of color h_0 in cluster $\bar{C}_i$
  Set $Q_i = \{j_{h_0}\}$.
  Update $s_{h_0} = s_{h_0} + 1$.
end for
while $\exists h \in \mathcal{H} : s_h < k_h^l$ **do**
  Pick a color h_0 such that $s_{h_0} < k_{h_0}^l$.
  Pick a center $i \in \bar{S}$ with cluster $\bar{C}_i$ where there exists a point of color h_0 not in Q_i.
  Pick a point j_{h_0} of color h_0 in cluster $\bar{C}_i$
  Set $Q_i = Q_i \cup \{j_{h_0}\}$.
  Update $s_{h_0} = s_{h_0} + 1$.
end while
Set $S = \cup_{i \in \bar{S}} Q_i$.
for $i \in \bar{S}$ **do**
  $\phi_i =$ DIVIDE$(\bar{C}_i, Q_i)$.
  $\forall j \in \bar{C}_i$: Set $\phi(j) = \phi_i(j)$. {Assignment to center is updated using DIVIDE.}
end for

Here we start with a solution $(\bar{S}, \bar{\phi})$ of cost $\bar{R}$ that satisfies the **GF** constraints and we want to make it satisfy **GF** and **DS** simultaneously. More specifically, given any **GF** solution we show how it can be post-processed to satisfy **GF+DS** at a bounded increase to its clustering cost by a

factor of 2 (see Theorem 2.5.8). This implies as a corollary that if we have an $\alpha_{\mathbf{GF}}$-approximation algorithm for **GF** then we can obtain a $2\alpha_{\mathbf{GF}}$-approximation algorithm for **GF+DS** (see Corollary 2.5.10).

The algorithm essentially first "covers" each given cluster $\bar{C}_i$ of the given solution $(\bar{S}, \bar{\phi})$ by picking a point of some color h to be a *future* center given that picking a point of such a color would not violate the **DS** constraints (lines(4-13)). If there are still colors which do not have enough picked centers (below the lower bound k_h^l), then more points are picked from clusters where points of such colors exist (lines(14-20)). Once the algorithm has picked correct points for each color, then the DIVIDE subroutine is called to divide the cluster among the picked points.

Now we state the main theorem:

Theorem 2.5.8. *If we have a solution $(\bar{S}, \bar{\phi})$ of cost $\bar{R}$ that satisfies the **GF** constraints where the number of non-empty clusters is $|\bar{S}| = \bar{k} \leq k$, then we can obtain a solution (S, ϕ) that satisfies **GF** at an additive violation of 2 and **DS** simultaneously with cost $R \leq 2\bar{R}$.*

Proof. We point out the following fact:

Fact 3. *Every cluster in $(\bar{S}, \bar{\phi})$ has at least one point from each color.*

Proof. This holds, since given a center $i \in \bar{S}$ we have $|\bar{C}_i| > 0$ and therefore $\forall h \in \mathcal{H} : |\bar{C}_i^h| \geq \beta_h |\bar{C}_i| > 0$ and therefore $|\bar{C}_i^h| \geq 1$ since it must be an integer. □

We note that the values $\{\beta_h, \alpha_h\}_{h \in \mathcal{H}}$ and k must lead to a feasible **DS** problem, i.e. there exist positive integers g_h such that $\sum_{h \in \mathcal{H}} g_h = k$ and $\forall h \in \mathcal{H} : \beta_h k \leq g_h \leq \alpha_h k$. Accordingly, since lines (4-13) in algorithm 11 can always pick a point of some color h such that the upper bound $\alpha_h k$ is not exceeded for every cluster i. Therefore the following fact must hold

Fact 4. *By the end of line (13) we have* $\forall i \in \bar{S} : |Q_i| \geq 1$.

Further, the final s_h values are valid for **DS**:

Claim 2.5.9. *By the end of line (13) the values of* s_h *satisfy: (1)* $\sum_{h \in \mathcal{H}} s_h \leq k$*, (2)* $\forall h \in \mathcal{H} : \beta_h k \leq s_h \leq \alpha_h k$.

Proof. Lines (4-13) add values to s_h if the lower bound $\beta_h k$ for color h is not satisfied. If the lower bound is satisfied for all colors, then points of some color h are added provided that adding them would not exceed the upper bound of $\alpha_h k$ (see line 5). Therefore, by the end of line (13) for any color $h \in \mathcal{H} : s_h \leq \alpha_h k$ and either $s_h \geq \beta_h k$ or $s_h < \beta_h k$[13].

If by the end of line (13) we have $\forall h \in \mathcal{H} : s_h \geq \beta_h k$, then the algorithm moves to line (22). Otherwise, it will keep picking points and incrementing s_h until $\forall h \in \mathcal{H} : s_h \geq \beta_h k$.

Further, since such valid **DS** values exist it must be that the above satisfies $\sum_{h \in \mathcal{H}} s_h \leq k$ and $\forall h \in \mathcal{H} : s_h \leq \alpha_h k$. This concludes the proof for the claim. □

By Lemma 2.5.2 for DIVIDE the new centers $S = \cup_{i \in \bar{S}} Q_i$ are all active (guarantee 2 of DIVIDE) and since the values of s_h are valid (Claim 2.5.9 above), therefore S satisfies the **DS** constraints.

Since the assignment in each cluster in the new solution (S, ϕ) is formed using DIVIDE over the clusters of $(\bar{S}, \bar{\phi})$ then by guarantee 1 of DIVIDE, each cluster (S, ϕ) satisfies **GF** at an additive violation of 2. Finally, the clustering cost is at most $R \leq 2\bar{R}$ (guarantee 3 of DIVIDE). □

Corollary 2.5.10. *Given an* $\alpha_{\mathbf{GF}}$*-approximation algorithm for* ***GF****, then we can have a* $2\alpha_{\mathbf{GF}}$*-approximation algorithm that satisfies* ***GF*** *at an additive violation of* 2 *and* ***DS*** *simultaneously.*

[13] To see why we could have $s_h < \beta_h k$, consider the case where $\bar{k} < k$ and therefore there would not be enough clusters to so that we can add points for each color.

Proof. Using the previous theorem (Theorem 2.5.8) the solution $(\bar{S}, \bar{\phi})$ has a cost of $\bar{R} \leq \alpha_{\mathbf{GF}} \operatorname{OPT}_{\mathbf{GF}}$. The post-processed solution that satisfies **GF** at an additive violation of 2 and **DS** simultaneously has a cost of $R \leq 2\bar{R} \leq 2\alpha_{\mathbf{GF}} \operatorname{OPT}_{\mathbf{GF}} \leq 2\alpha_{\mathbf{GF}} \operatorname{OPT}_{\mathbf{GF+DS}}$. The last inequality follows because $\operatorname{OPT}_{\mathbf{GF}} \leq \operatorname{OPT}_{\mathbf{GF+DS}}$ which is the case since both problems minimize the same objective, however by definition the constraint set of **GF** + **DS** is a subset of the constraint set of **GF**. □

Remark: If the given **GF** solution has the number of cluster $\bar{k} = k$, then the output will have an additive violation of zero, i.e. satisfy the **GF** constraints exactly. This would happen DIVIDE would always receive Q_i with $|Q_i| = 1$ and therefore we can use the guarantee of DIVIDE for the special case of $|Q| = 1$.

2.5.4 Price of (Doubly) Fair Clustering

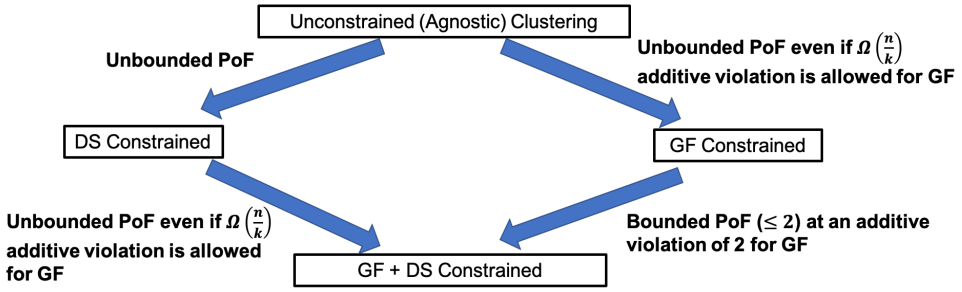


Figure 2.21: Figure showing the PoF relation between Unconstrained, **GF**, **DS**, and **GF+DS** clustering.

Here we study the degradation in the clustering cost (the price of fairness) that comes from imposing the fairness constraint on the clustering objective. The price of fairness PoF_c is defined as $\operatorname{PoF}_c = \frac{\text{Clustering Cost subject to Constraint } c}{\text{Clustering Cost of Agnostic Solution}}$ [17, 40]. Note that since we have two constrains here **GF** and **DS**, we also consider prices of fairness of the form $\operatorname{PoF}_{c_1 \to c_2} = \frac{\text{Clustering Cost subject to Constraints } c_1 \text{ and } c_2}{\text{Clustering Cost subject to Constraint } c_1}$ which equal the amount of degradation in the clustering cost if we were to impose constraint c_2

in addition to constraint c_1 which is already imposed. Note that we are concerned with the price of fairness in the *worst case*. Interesingly, we find that imposing the **DS** constraint over the **GF** constraint leads to a bounded PoF if we allow an additive violation of 2 for **GF** while the reverse is not true even if we allow an additive violation of $\Omega(\frac{n}{k})$ for **GF**.

We find the following:

Proposition 2.5.11. *For any value of $k \geq 2$, imposing **GF** can lead to an unbounded* PoF *even if we allow an additive violation of $\Omega(\frac{n}{k})$.*

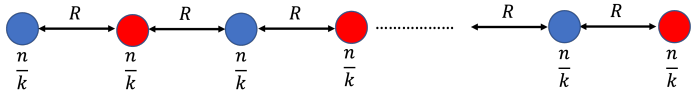


Figure 2.22: An clustering instance made of k "masses" each having $\frac{n}{k}$ points. Each mass is seperated from the other by a distance of at least R.

Proof. Consider the case where $k \geq 2$ is even and refer to Figure 2.22 where the optimal solution has a clustering cost of 0. The optimal solution would have one center in each of the k masses, and assigns points to their closest center.

If we set the lower and upper proportion bounds to $\frac{1}{2}$ for both colors, then to satisfy **GF** each cluster should have both red and blue points. There must exists a cluster C_i of size $|C_i| \geq \frac{n}{k}$, it follows that to satisfy the **GF** constraints at an additive violation of ρ, then $|C_i^{\text{blue}}| \geq \frac{1}{2}|C_i| - \rho = \frac{n}{2k} - \rho$ and similarly we would have $|C_i^{\text{red}}| \geq \frac{n}{2k} - \rho$. By setting $\rho = \frac{n}{2k} - \epsilon$ for some constant $\epsilon > 0$, then we have $|C_i^{\text{blue}}|, |C_i^{\text{red}}| > 0$. This implies that a point will be assigned to a center at a distance $R > 0$ and therefore the PoF is unbounded.

The proof for the odd case follows by constructing a similar example and argument. □

Proposition 2.5.12. *For any value of $k \geq 3$, imposing **DS** can lead to an unbounded* PoF.

Figure 2.23: This clustering instance is similar to Figure 2.22 except that the color assignments follow a different pattern.

Proof. Consider the example shown in Figure 2.23 where $k \geq 3$ and $k = \ell$. Here all masses are blue, except for the last which has $\frac{n}{2k}$ red points and $\frac{n}{2k}$ green points.

Suppose for **DS** we set $k^l_{\text{blue}}, k^l_{\text{red}}, k^l_{\text{green}} > 0$, this implies that we should pick a center of each color. This implies that we can have at most $k - 2$ blue center, therefore there will be a community (composed of all blue points) where no point is picked as a center. Therefore, the clustering cost is $R > 0$ and the PoF is unbounded. □

Proposition 2.5.13. *For any value of $k \geq 2$, imposing **GF** on a solution that only satisfies **DS** can lead to an unbounded increase in the clustering cost even if we allow an additive violation of $\Omega(\frac{n}{k})$.*

Proof. The proof is similar to the proof of Proposition 2.5.11. See [24] for the full proof. □

Proposition 2.5.14. *Imposing **DS** on a solution that only satisfies **GF** leads to a bounded increase in the clustering cost of at most 2 (PoF ≤ 2) if we allow an additive violation of 2 in the **GF** constraints.*

Proof. This follows from Theorem 2.5.8 since we can always post-process a solution that only satisfies **GF** into one that satisfies both **GF** at an additive violation of 2 and **DS** simultaneously and clearly from the theorem we would have $\text{PoF} = \frac{\text{clustering cost of } \textbf{GF} \text{ post-processed solution}}{\text{clustering cost of } \textbf{GF} \text{ solution}} \leq \frac{2 \text{ clustering cost of } \textbf{GF} \text{ solution}}{\text{clustering cost of } \textbf{GF} \text{ solution}} \leq 2.$ □

2.5.5 Incompatibility with Other Distance-Based Fairness Constraints

In this section, we study the incompatibility between the **DS** and **GF** constraints and a family of distance-based fairness constraints. We note that the results of this section do not take into account the clustering cost and are based only on the feasibility set. That is, we consider more than one constraints simultaneously and see if the feasibility set is empty or not. Two constraints are considered *incompatible* if the intersection of their feasible sets is empty. In some cases we also consider solution that could have violations to the constraints. Note that socially fair clustering [77,78] is defined as an optimization problem not a constraint. However, by setting a constraint that says the value of the objective function should not exceed an upper bound it can be turned into a fairness constraint like the rest.

Theorem 2.5.15. *For any value $k \geq 2$, the **fairness in your neighborhood** [79], **socially fair** constraint [77, 78] are each incompatible with **GF** even if we allow an additive violation of $\Omega(\frac{n}{k})$ in the **GF** constraint. For any value $k \geq 5$, the **proportionally fair** constraints [80] is incompatible with **GF** even if we allow an additive violation of $\Omega(\frac{n}{k})$ in the **GF** constraint.*

Proof. The proofs for the **fairness in your neighborhood** and **socially fair** constraints are very similar to the proof of Proposition 2.5.11. However, the **proportionally fair** constraint requires a different proof.

Suppose, we want an α_{AP} relaxed proportionally fair solution [80]. Then for a given finite value of α_{AP}, consider Figure 2.24. For the **GF** constraints, the upper and lower bounds for each color to $\frac{1}{2}$ and the total number of points n is always even. Consider some $k \geq 5$. It follows that the sum of cluster sizes assigned to centers on either the right side or the left side would be at least $\frac{n}{2}$, WLOG assume that it is the left side and denote the total number of points assigned to

clusters on the left size by $|C_{LS}|$ and let S_{LS} be the centers on the left side. The total number of points on the left side may not be assigned to a single center but rather distributed among the centers S_{LS}. To satisfy the **GF** constraints at an additive violation of ρ, it follows that the number of red points that have to be assigned to the left side is at least $\sum_{i \in S_{LS}} (\frac{1}{2}|C_i| - \rho) \geq \frac{n}{4} - k\rho$. Set $\rho = \frac{n}{4k} - \frac{n}{k^2} - 1$, then it follows that at least $\lceil \frac{n}{k} \rceil$ red points are assigned to a center on the left at a distance of at least R. Since the maximum distance between any two red points by the triangle inequality is $2r < \frac{R}{\alpha_{AP}}$ it follows that this set of red points forms a blocking coalition. I.e., these points would also have a lower distance from their assigned center if they were instead assigned to a red center. □

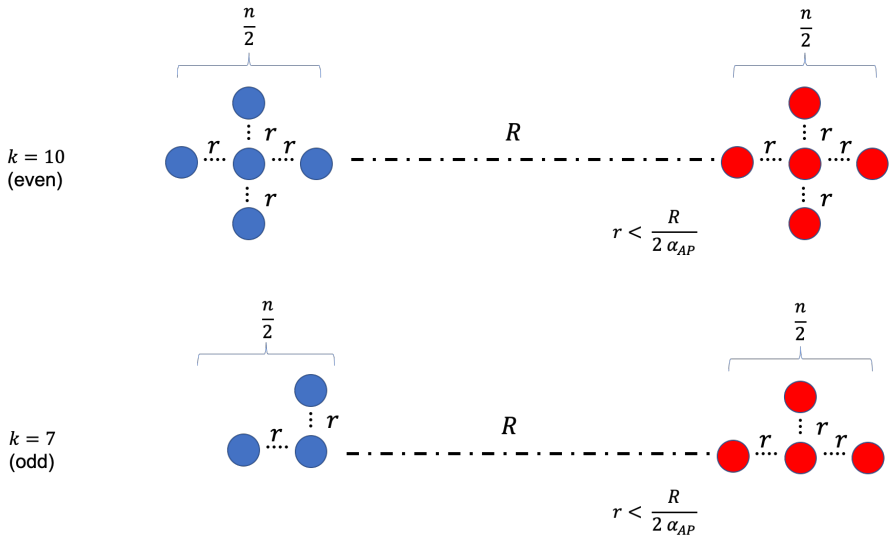


Figure 2.24: Instances to show incompatibility between Proportional Fairness and **GF**. We always have $n/2$ blue points on the left and $n/2$ red points on the right. For even k we would have $k/2$ locations for the blue and red points each. For odd k we have $\lfloor k/2 \rfloor$ blue locations and $\lceil k/2 \rceil$ red locations. For each color, there is always a location at the center at a distance r from the other locations. Points of different color are at a distance of at least R from each other. For any value of α_{AP} for the proportionally fair constraint, we set $r < \frac{R}{2\alpha_{AP}}$.

Theorem 2.5.16. *For any value $k \geq 3$, the **fairness in your neighborhood** [79], **socially fair** [77, 78] and **proportionally fair** [80] constraints are each incompatible with **DS**.*

Proof. The proof is similar to the proof of Proposition 2.5.12. Specifically, the example in the proof only allows solutions where all of the points in each mass are assigned to centers in the same mass to be feasible for any of the distance based fairness constraints. But since such a solution would not satisfy **DS**, then there is no feasible solution. □

Compatibility between GF and DS: One can easily show compatibility between **GF** and **DS**. Specifically, consider some values for the centers over the colors $\{k_h\}_{h\in\mathcal{H}}$ that satisfies the **DS** constraints, i.e. $\forall h \in \mathcal{H} : k_h^l \leq k_h \leq k_h^u$ and has $\sum_{h\in\mathcal{H}} k_h \leq k$. Then simply pick a set Q_h of k_h points of color h. Now if we give DIVIDE the entire dataset $\mathcal{C}$ and the set of centers $\cup_{h\in\mathcal{H}} Q_h$ as inputs, i.e. call DIVIDE$(\mathcal{C}, \cup_{h\in\mathcal{H}} Q_h)$, then by the guarantees of divide each center would be active and each cluster would satisfy the **GF** constraints at an additive violation of 2.

Our final conclusions about the incompatibility and compatibility of the constrains are summarized in Figure 2.25.

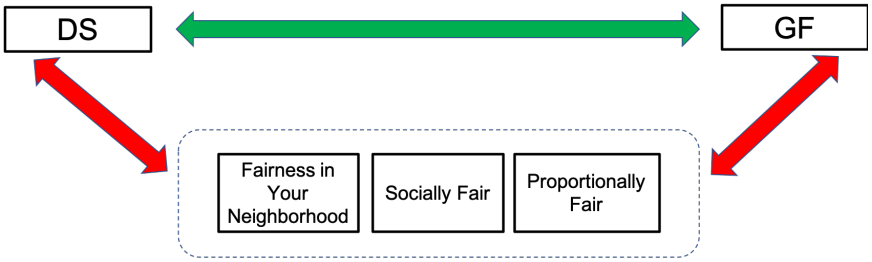


Figure 2.25: (In)Compatibility of clustering constraints. Red arrows indicate empty feasible set when both constraints are applied, while green arrows indicate non-empty feasibility set when both constraints are applied.

2.5.6 Experiments

We use Python 3.9, the `CPLEX` package [81] for solving linear programs and `NetworkX` [82] for max-flow rounding. Further, `Scikit-learn` is used for some standard ML related operations. We use commodity hardware, specifically a MacBook Pro with an Apple M2 chip.

We conduct experiments over datasets from the UCI repository [83] to validate our theoretical findings. Specifically, we use the **Adult** dataset sub-sampled to 20,000 records. Gender is used for group membership while the numeric entries are used to form a point (vector) for each record. We use the Euclidean distance. Further, for the **GF** constraints we set the lower and upper proportion bounds to $\beta_h = (1-\delta)r_h$ and $\alpha_h = (1+\delta)r_h$ for each color h where r_h is color h's proportion in the dataset and we set $\delta = 0.2$. For the **DS** constraints, since we do not deal with a large number of centers we set $k_h^l = 0.8 r_h k$ and $k_h^u = r_h k$.

We compare the performance of 5 algorithms. Specifically, we have (1) **COLOR-BLIND:** An implementation of the Gonzalez k-center algorithm [41] which achieves a 2-approximation for the unconstrained k-center problem. (2) **ALG-GF:** A **GF** algorithm which follows the sketch of [40], however the final rounding step is replaced by an implementation of the MAXFLOWGF rounding subroutine. This algorithm has a 3-approximation for the **GF** constrained instance. (3) **ALG-DS:** An algorithm for the **DS** problem recently introduced by [23] for which also has an approximation of 3. (4) **GFTOGFDS:** An implementation of algorithm 11 where we simply use the **GF** algorithm just mentioned to obtain a **GF** solution. (5) **DSTOGFDS:** Similarly an implementation of algorithm 9 where **DS** algorithm is used as a starting point instead.

Throughout we measure the performance of the algorithms in terms of (1) **PoF:** The price of fairness of the algorithm. Note that we always calculate the price of fairness by dividing by

the COLOR-BLIND clustering cost since it solves the unconstrained problem. (2) **GF-Violation:** Which is the maximum additive violation of the solution for the **GF** constraint as mentioned before. (3) **DS-Violation:** Which is simply the maximum value of the under-representation or over-representation across all groups in the selected centers.

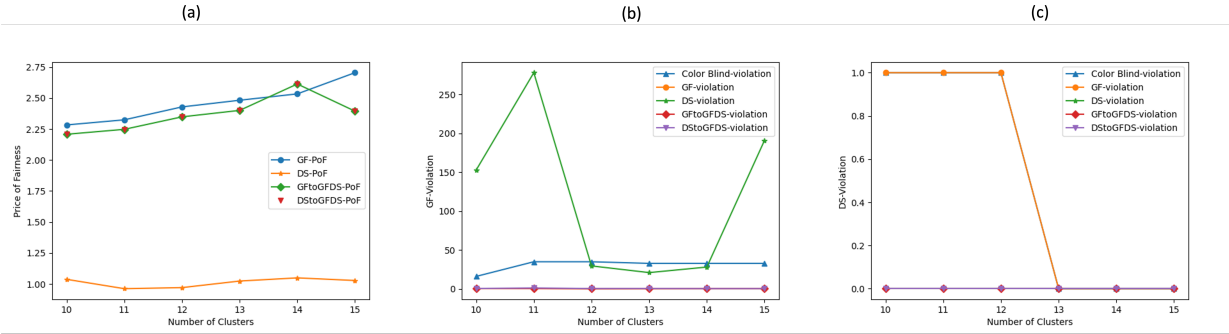


Figure 2.26: **Adult** dataset results: (a) **PoF** comparison of 5 algorithms, with **COLOR-BLIND** as baseline; (b) **GF-Violation** comparison; (c) **DS-Violation** comparison.

Figure 2.26 shows the behaviour of all 5 algorithms. In terms of **PoF**, all algorithms have a significant degredation in the clustering cost compared to the COLOR-BLIND baseline except for ALG-DS. However, ALG-DS has a very large **GF-Violation**. In fact, the **GF-Violation** of ALG-DS can be more than 5 times the **GF-Violation** of COLOR-BLIND. This indicates that while ALG-DS has a small clustering cost, it can give very bad guarantees for the **GF** constraints. Finally, in terms of the **DS-Violation** we see that the ALG-GF and the COLOR-BLIND solution can violate the **DS** constraint. Note that both coincide perfectly on each other. Further, although the violation is 1, it is very significant since unlike the **GF** constraints the number of centers can be very small. On the other hand, we see that both GFTOGFDS and DSTOGFDS give the best of both worlds having small values for the **GF-Violation** and zero values for the **DS-Violation** and while their price of fairness can be significant, it is comparable to ALG-GF. Interestingly, the GFTOGFDS and DSTOGFDS are in agreement in terms of measures. This could be because

our implementations of the "**GF** part" of DSTOGFDS (its handling of the **GF** constraints) has similarities to the GFTOGFDS algorithm.

Chapter 3: Fairness in Online Bipartite Matching

Online bipartite matching has been used to model many important applications such as crowdsourcing [84–86], rideshare [87–89], and online ad allocation [26, 90]. 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, [91–93] report that drivers could reject riders based on attributes such as gender, race,

or disability. Conversely, [94] reports that drivers are likely to receive less favorable ratings if they belong to certain demographic groups. A similar phenomenon exists in crowdsourcing [95]. 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 [96, 97], 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. [98–100] present algorithms which give a minimum utility guarantee for the drivers at a bounded drop to the operator's profit. Conversely, [101] give guarantees for both the platform operator and the riders instead. Finally, [102] 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 **KIID** and the richer **KAD** setting (see Section 3.2 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 **KAD** 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.

3.1 Related Work

It is worth noting that similar to our work, [103] and [104] have considered two-sided fairness as well, although in the setting of recommendation systems where a different model is applied—and, critically, a separate objective for the operator's profit was not considered.

Fairness in bipartite matching has seen significant interest recently. The fairness definition employed has consistently been the well-known Rawlsian fairness [105] (i.e. max-min fairness) or its generalization Leximin fairness.[1] We note that the objective to be maximized (other than the fairness objective) represents operator profit in our setting.

The case of offline and unweighted maximum cardinality matching is addressed by [106], who give an algorithm with Leximin fairness guarantees for one side of the market (one side of the bipartite graph) and show that this can be achieved without sacrificing the size of the match. Motivated by fairness consideration for drivers in ridesharing, [98] considers the problem of offline and weighted matching. Specifically, they show an algorithm with a provable trade-off between the operator's profit and the minimum utility guaranteed to any vertex in one-side of the market.

Recently, [29] considered fairness for the online part of the graph through a group notion of fairness. In particular, the utility for a group is added across the different types and is minimized for the group worst off, in rough terms their notion translates to maximizing the minimum utility

[1] Leximin fairness maximizes the minimum utility like max-min fairness. However, it proceeds to maximize the second worst utility, and so on until the list is exhausted.

accumulated by a group throughout the matching. Their notion of fairness is very similar to the one we consider here. However, [29] considers fairness only on one side of the graph and ignores the operator's profit. Further, only the matching size is considered to measure utility, i.e. edges are unweighted.

A new notion of group fairness in online matching is considered in [107]. In rough terms, their group fairness criterion amounts to establishing a quota for each group and ensuring that the matching does not exceed that quota. This notion can be seen as ensuring that the system is not dominated by a specific group and is in some sense an opposite to max-min fairness as the utility is upper bounded instead of being lower bounded. Further, the fairness guarantees considered are one-sided as well.

On the empirical side of fair online matching, [108] and [109] give application-specific treatments in the context of deceased-donor organ allocation and food bank provisioning, respectively. More related to our work is that of [102, 110] which consider the rideshare problem and provide algorithms to achieve fairness for both sides of the graph simultaneously, however both papers lack theoretical guarantees and in the case of [102] the operator's profit is not considered.

3.2 Preliminaries and Problem Setup

Our model follows that of [26, 111–113] 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 dynamically in each round. 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 (**KAD**) setting [88, 113]. 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 (**KIID**). 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 **KIID** setting. We assume that $n_v \in \mathbb{Z}^+$ for **KIID** [112]. Every vertex u (v) has a group membership,[2] with $\mathcal{G}$ 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 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

[2]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.

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.

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:**

  - **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

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

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

  - **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:**

  - **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 $\min_{u \in U} \mathbb{E}[\sum_{e \in \mathcal{M}_u} w_e^U]$.

  - **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.

3.3 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 **KIID** arrival setting we have:

Theorem 3.3.1. *For the **KIID** setting, algorithm* $\mathrm{TSGF}_{\boldsymbol{KIID}}(\alpha, \beta, \gamma)$ *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 **KAD** setting we have:

Theorem 3.3.2. *For the **KAD** setting, algorithm* $\mathrm{TSGF}_{\boldsymbol{KAD}}(\alpha, \beta, \gamma)$ *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 **KIID** or **KAD** arrival setting we have:

Theorem 3.3.3. *For the **KIID** or **KAD** 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* $\alpha + \beta + \gamma \leq 1$.

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

Theorem 3.3.4. *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* $\alpha + \beta + \gamma > 1$.

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

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 3.3.5. *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 3.6.

3.4 Algorithms and Theoretical Guarantees

Our algorithms use linear programming (LP) based techniques [100, 101, 112, 114] 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.

3.4.1 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 simply "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 **KIID** 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 (3.4), but differing by the objective. For compactness we do not repeat these constraints and instead write them once. Specifically, LP objective (3.1) along with the constraints of (3.4) give the optimal benchmark value of the operator's profit. Similarly, with the same set of constraints (3.4) LP objective (3.2) and LP objective (3.3) give the optimal group max-min fair assignment for the offline and online sides, respectively. Note that the expected max-min objectives of (3.2) and (3.3), can be written in the form of a linear objective. For example, the max-min objective of (3.2) 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(3.1), LP(3.2), and LP(3.3) to refer to the platform's profit LP, the offline side group fairness LP, and the online side group fairness LP, respectively.

$$\max \sum_{e \in E} w_e^O x_e p_e \tag{3.1}$$

$$\max \min_{g \in \mathcal{G}} \frac{\sum_{u \in U(g)} \sum_{e \in E_u} w_e^U x_e p_e}{|U(g)|} \tag{3.2}$$

$$\max \min_{g \in \mathcal{G}} \frac{\sum_{v \in V(g)} \sum_{e \in E_v} w_e^V x_e p_e}{|V(g)|} \tag{3.3}$$

$$\text{s.t} \quad \forall e \in E : 0 \leq x_e \leq 1 \tag{3.4a}$$

$$\forall u \in U : \sum_{e \in E_u} x_e p_e \leq 1 \tag{3.4b}$$

$$\forall u \in U : \sum_{e \in E_u} x_e \leq \Delta_u \tag{3.4c}$$

$$\forall v \in V : \sum_{e \in E_v} x_e p_e \leq 1 \tag{3.4d}$$

$$\forall v \in V : \sum_{e \in E_v} x_e \leq \Delta_v \tag{3.4e}$$

Now we prove that LP(3.1), LP(3.2) and LP(3.3) 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 3.4.1. *For the **KIID** setting, the optimal solutions of LP (3.1), LP (3.2), and LP (3.3) 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 [112]. We shall focus on the operator's profit objective since the other objectives follow by very similar arguments. First, we note that LP(3.1) 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(3.1) 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(3.1) 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:

$$\max \sum_{e' \in E'} w^O_{e'} p_{e'} x_{e'} \tag{3.5}$$

$$\text{s.t} \quad \forall e' \in E' : 0 \leq x_{e'} \leq 1 \tag{3.6a}$$

$$\forall u \in U : \sum_{e' \in E'_u} x_{e'} p_{e'} \leq 1 \tag{3.6b}$$

$$\forall u \in U : \sum_{e' \in E'_u} x_{e'} \leq \Delta_u \tag{3.6c}$$

$$\forall v' \in V' : \sum_{e' \in E'_{v'}} x_{e'} p_{e'} \leq 1 \tag{3.6d}$$

$$\forall v' \in V' : \sum_{e' \in E'_{v'}} x_{e'} \leq \Delta_{v'} \tag{3.6e}$$

where V' is the realization of the online side. It is clear that for a given realization $G' = (U, V', E')$ the above LP(3.5) 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:

$$\min \sum_{u \in U} (\alpha_u + \Delta_u \beta_u) + \sum_{v' \in V'} (\alpha_{v'} + \Delta_{v'} \beta_{v'}) + \sum_{(u,v')} \gamma_{u,v'} \tag{3.7}$$

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

$$\beta_u + \beta_{v'} + p_{(u,v')}(\alpha_u + \alpha_{v'}) + \gamma_{(u,v')} \geq w^O_{(u,v')} p_{(u,v')} \tag{3.8a}$$

$$\alpha_u, \alpha_{v'}, \beta_u, \beta_{v'}, \gamma_{(u,v')} \geq 0 \tag{3.8b}$$

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 3.8a and 3.8b. 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 $n_v^{G'}$, then the value of the solution $\vec{\hat{\alpha}}, \vec{\hat{\beta}}, \vec{\hat{\gamma}}$ satisfies:

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

Now taking the expectation, we get:

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

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 [69]. 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 (3.1,3.2,3.3), we denote the optimal solutions to LP (3.1), LP (3.2), and LP (3.3) 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 12.

Algorithm 12 PPDR$(\vec{x}_v)$

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

The procedure of our parameterized sampling algorithm $\text{TSGF}_{\textbf{KIID}}$ is shown in Algorithm 13. Specifically, when a vertex of type v arrives at any time step we run **PPDR**$\left(\vec{x}_v^*\right)$, **PPDR**$\left(\vec{y}_v^*\right)$, or **PPDR**$\left(\vec{z}_v^*\right)$ 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 (3.4e) 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 13 $\text{TSGF}_{\textbf{KIID}}(\alpha, \beta, \gamma)$

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

Now we analyze $\text{TSGF}_{\textbf{KIID}}$ to prove Theorem 3.3.1. 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 $SF_{u,t}$.

Lemma 3.4.2. $Pr[SF_{u,t}] \geq \left(1 - \frac{t-1}{T}\right)\left(1 - \frac{1}{T}\right)^{t-1}$.

Proof. We have to first introduce the following two claims. Specifically, let $A_{u,t}$ be the number of successful assignments that u received and accepted before round t. Then the following claim holds.

Claim 3.4.3. *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 $\text{TSGF}_{\textbf{KIID}}(\alpha, \beta, \gamma)$ algorithm at time $k < t$. Let $Z_{e,k}$ be the indicator random variable that assign-

ment $e = (u, v)$ 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 claim holds:

Claim 3.4.4. $Pr[B_{u,t} < \Delta_u] \geq 1 - \frac{t-1}{T}$.

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

Now we are ready to prove the lemma, consider a given edge $e \in E_u$ where $k < t$

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

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

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

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:

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

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,

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

□

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 3.4.5). Let $1_{e,t}$ be the indicator that $e = (u, v)$ is probed by the $\text{TSGF}_{\textbf{KIID}}$ 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 3.4.5. $Pr[1_{e,t} \mid SF_{u,t}] \geq \alpha \frac{x_e^*}{2T}$,$Pr[1_{e,t} \mid SF_{u,t}] \geq \beta \frac{y_e^*}{2T}$, $Pr[1_{e,t} \mid SF_{u,t}] \geq \gamma \frac{z_e^*}{2T}$

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 13 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 12) 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:

$$Pr[1_{e,t} \mid SF_{u,t}] \tag{3.9}$$

$$\geq \alpha \frac{1}{T} Pr[X_e = 1] Pr\big[\sum_{e' \in E_{r,\bar{e}}} X_{e'} Y_{e'} Z_{e'} \mid X_e = 1\big] \tag{3.10}$$

$$= \alpha \frac{Pr[X_e = 1]}{T} \big(1 - Pr\big[\sum_{e' \in E_{v,\bar{e}}} X_{e'} Y_{e'} Z_{e'} \geq 1 \mid X_e = 1\big]\big) \tag{3.11}$$

$$\geq \alpha \frac{Pr[X_e = 1]}{T} \big(1 - \mathbb{E}\big[\sum_{e' \in E_{v,\bar{e}}} X_{e'} Y_{e'} Z_{e'} \geq 1 \mid X_e = 1\big]\big) \tag{3.12}$$

$$\geq \alpha \frac{Pr[X_e = 1]}{T} \big(1 - \sum_{e' \in E_{v,\bar{e}}} \mathbb{E}\big[X_{e'} Y_{e'} Z_{e'} \geq 1 \mid X_e = 1\big]\big) \tag{3.13}$$

$$\geq \alpha \frac{x^*_e}{T} \big(1 - \sum_{e' \in E_{v,\bar{e}}} x^*_{e'} \frac{1}{2} p_{e'}\big) \tag{3.14}$$

$$\geq \alpha \frac{x^*_e}{T} \big(1 - \frac{1}{2}\big) = \alpha \frac{x^*_e}{2T} \tag{3.15}$$

Applying Markov inequality we get the inequality (3.12). By linearity of expectation we get inequality (3.13). 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 (3.14). The last inequality (3.15) 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 3.3.1 can be proved.

Proof of Theorem 3.3.1. Denote the expected number of probes on each edge $e \in E$ resulting

from **PPDR**$\big(\vec{x}^*_v\big)$ by n^x_e. It follows that:

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

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: $ALG_O = \mathbb{E}\Big[\sum_{e\in E} w^O_e n^x_e p_e\Big] \geq \sum_{e\in E} w^O_e p_e \frac{\alpha x^*_e}{2e} \geq \sum_{e\in E} w^O_e p_e \Big(\frac{1}{e}\Big)\frac{\alpha x^*_e}{2} \geq \frac{\alpha}{2e}(OPT_O)$. 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. □

3.4.2 Group Fairness for the **KAD** Setting:

For the **KAD** 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 **KAD** 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:

$$\max \sum_{t\in[T]} \sum_{e\in E} w_e^O x_{e,t} \tag{3.16}$$

$$\max \min_{g\in\mathcal{G}} \frac{\sum_{t\in[T]} \sum_{u\in U(g)} \sum_{e\in E_u} w_e^U x_{e,t}}{|U(g)|} \tag{3.17}$$

$$\max \min_{g\in\mathcal{G}} \frac{\sum_{t\in[T]} \sum_{v\in V(g)} \sum_{e\in E_v} w_e^V x_{e,t}}{\sum_{v\in V(g)} n_v} \tag{3.18}$$

$$\text{s.t} \quad \forall e \in E, \forall t \in [T] : 0 \leq x_{e,t} \leq 1 \tag{3.19a}$$

$$\forall u \in U : \sum_{t\in[T]} \sum_{e\in E_u} x_{e,t} \leq 1 \tag{3.19b}$$

$$\forall v \in V, \forall t \in [T] : \textstyle\sum_{e\in E_v} x_{e,t} \leq p_{v,t} \tag{3.19c}$$

Lemma 3.4.6. *For the **KAD** setting, the optimal solutions of LP (3.16), LP (3.17) and LP (3.18) 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:

$$\max \sum_{t\in[T]} \sum_{e\in E} w_e^O x_{e,t} \tag{3.20}$$

$$\text{s.t} \quad \forall e \in E, \forall t \in [T] : 0 \leq x_{e,t} \leq 1 \tag{3.21a}$$

$$\forall u \in U : \sum_{t \in [T]} \sum_{e \in E_u} x_{e,t} \leq 1 \tag{3.21b}$$

$$\forall v \in V, \forall t \in [T] : \textstyle\sum_{e \in E_v} x_{e,t} \leq 1_{v,t} \tag{3.21c}$$

If we were to replace the random variables $1_{v,t}$ by their expected value, then we would retrieve LP(3.16) where $\mathbb{E}_{\mathcal{I}}[1_{v,t}] = p_{v,t}$. It suffices to show that the value of LP(3.16) 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 $x^{*,G}_{e,t}$ 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(3.16), 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 $\mathbb{E}_{\mathcal{I}}[LP(G)] = \mathbb{E}_{\mathcal{I}}[\sum_{t \in [T]} \sum_{e \in E} w^O_e x^{*,G}_{e,t}] = \sum_{t \in [T]} \sum_{e \in E} w^O_e \mathbb{E}_{\mathcal{I}}[x^{*,G}_{e,t}] \leq \sum_{t \in [T]} \sum_{e \in E} w^O_e x^*_{e,t} = LP(\mathbb{E}_{\mathcal{I}}[G])$ where $x^*_{e,t}$ is the optimal solution for LP(3.16) 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 **KIID** 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 **KIID** 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 $\text{TSGF}_{\textbf{KAD}}$ shown in Algorithm 14. In $\text{TSGF}_{\textbf{KAD}}$ 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 (3.19c) 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 (3.19c) 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 3.4.7. *Algorithm* $\text{TSGF}_{\textbf{KAD}}$ *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:

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

where we used the fact that $x^*_{e,t'}, y^*_{e,t'}, z^*_{e,t'} \leq 1$ from constraint (3.19a) 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 [88, 115]. To elaborate, we note that we first solve the LPs (3.16,3.17,3.18) 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 of times, and use the empirically estimated probability. More precisely,

Algorithm 14 $\text{TSGF}_{\textbf{KAD}}(\alpha, \beta, \gamma)$

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

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 [116] 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 [88] and hence for simplicity it is ignored. Now, with Lemma 3.4.7 Theorem 3.3.2 can be proved.

Proof of Theorem 3.3.2. 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. □

3.4.3 Individual Fairness **KIID** and **KAD** Settings:

For the case of Rawlsian (max-min) individual fairness, we simply consider each vertex of the offline side to belong to its own distinct group and the definition of group max-min fairness would simply 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 $\text{util}(v_t)$ 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 **KAD** arrival setting:

Lemma 3.4.8. *Whether in the **KIID** or **KAD** 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 **KAD** 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 $\text{TSGF}_{\textbf{KAD}}$ to the reduced instance leads to the following corollary:

Corollary 3.4.9. *Given an instance of two-sided individual max-min fairness, applying* $\text{TSGF}_{\textbf{KAD}}(\alpha, \beta, \gamma)$ *to the reduction from Theorem 3.4.8 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 3.3.3 is immediate from the above corollary.

3.5 Proofs of Impossibility Results

Here we prove Theorems 3.3.4 and 3.3.5.

Proof of Theorem 3.3.4. We prove it for group fairness in the **KIID** setting, since the **KIID** setting is a special case of the **KAD** setting, then this also proves the upper bound for the **KAD** 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. **KIID** 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:

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

in the above, 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 **KAD** 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. □

Proof of Theorem 3.3.5. 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 $\mathrm{OPT}^V_I = \min\{\theta_1, L\theta_2\}$ and by setting T to an arbitrarily large value $\mathrm{OPT}^V_G = L$. Then we

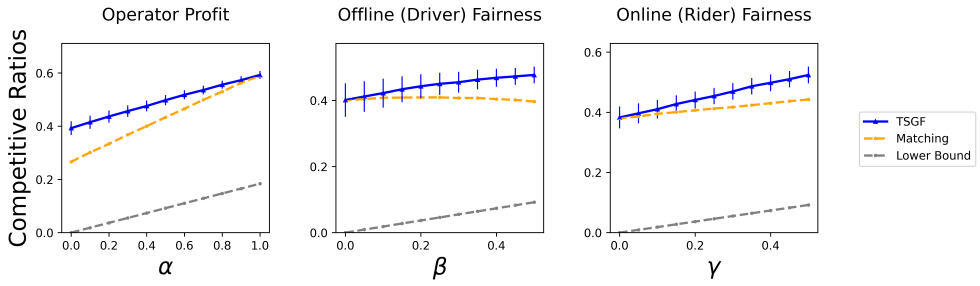


Figure 3.1: Competitive ratios for $\text{TSGF}_{\textbf{KIID}}$ over the operator's profit, offline (driver) fairness objective, and online (rider) fairness objective with different values of α, β, γ. Note that "Matching" refers to the case where driver and rider utilities are set to 1 across all edges. 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.

follow an identical argument to the above. □

3.6 Experiments

In this section, we verify the performance of our algorithm and our theoretical lower bounds for the KIID and group fairness setting using algorithm $\text{TSGF}_{\textbf{KIID}}$ (Section 3.4.1). 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., [88, 100, 101, 117]). 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 [100, 101, 118, 119] 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 [88, 101], 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 [100], 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 [101]. 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 [102] 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 $\text{TSGF}_{\textbf{KIID}}$ 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 $\text{TSGF}_{\textbf{KIID}}$ 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 3.1.

Performance of $\text{TSGF}_{\textbf{KIID}}$ with Varied Parameters: Figure 3.1 shows the performance of our algorithm over the three objectives: operator's profit, offline (driver) group fairness, and online (rider) group fairness. 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., [99–101]) 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 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 3.1, it is clear that ignoring the match quality leads to noticeably worse results.

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

Table 3.1: Competitive ratios of $\mathrm{TSGF}_{\mathbf{KIID}}$ with Greedy heuristics on the NYC dataset at $|U| = 49$, $|V| = 172$. Higher competitive ratio indicates better performance.

Comparison to Heuristics: We also compare the performance of $\mathrm{TSGF}_{\mathbf{KIID}}$ 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 $\mathrm{TSGF}_{\mathbf{KIID}}$ 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 below. We see that $\mathrm{TSGF}_{\mathbf{KIID}}$ 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 $\mathrm{TSGF}_{\mathbf{KIID}}$.

Chapter 4: Implications of Distance over Redistricting Maps: Central and Outlier Maps

In this chapter we consider redistricting which is a fundamental problem in any democracy. Redistricting is the process of dividing an electorate into a collection of districts which each elect a representative. In the United States, this process is used for both federal and state-level representation, and we will use the U.S. House of Representatives as a running example throughout this paper. Subject to both state and federal law, the division of states into congressional districts is not arbitrary and must satisfy a collection of properties such as districts being contiguous and of near-equal population. Despite these regulations, it is clear that redistricting is vulnerable to strategic manipulation in the form of gerrymandering. The body in charge of redistricting can easily create a map within the legal constraints that leads to election results which favor a particular outcome (e.g., more representatives elected from one political party in the case of *partisan gerrymandering*). In addition, the ability to draw gerrymandered districts has improved greatly with the aid of computers since the historic salamander-shaped district approved by Massachusetts Governor Elbridge Gerry in 1812. For example, assuming voting consistent with the 2016 election, the state of North Carolina with 13 representatives can be redistricted to elect either 3 Democrats and 10 republicans or 10 Democrats and 3 Republicans.

However, despite this obvious threat to functioning democracy, partisan gerrymandering

has often eluded regulation partly because it has been difficult to measure. In response, a recent line of research introduced sampling techniques to randomly[1] generate a large collection of redistricting maps [32, 120, 121] and calculate statistics such as a histogram of the number of seats won by each party using this collection. With these statistics, one can check if a proposed or enacted map is an outlier with respect to the sample. For example, the 2012 redistricting map of North Carolina produced 4 seats for the democratic party whereas 95% of the sampled maps led to between 6 and 9 seats [33]. In fact, these techniques were used as a key argument in the most recent U.S. Supreme Court case on gerrymandering [122] and have supported successful efforts to change redistricting maps in state supreme court cases [34]. More importantly for the present work, at least two states, Michigan and Wisconsin, will use such a sampling tool [32] in the current redistricting process in response to the 2020 U.S. Census [123].

While great progress has been made in recent years on the problem of detecting/labeling possible gerrymandering through the use of these sampling techniques which can quantify outlier characteristics in a given map, the question of drawing a redistricting map in a way that is "fair" and immune to strategic manipulation remains largely unclear. We survey some existing proposals to automate redistricting in more detail in Section 4.1, but none of them have been adopted in practice thus far. The direction most commonly proposed by automated redistricting methods is to cast redistricting as a constrained optimization problem [124–126], with objectives such as compactness and a collection of common constraints (district contiguity, equal population, etc.). However, formulating redistricting as an optimization problem poses an issue in the fact that there are multiple desired properties, and it is not clear why one should be optimized for over others.

[1] These are not the truly uniform random samples from the immense and ill-defined space of all possible maps that we ideally want, but they are generally treated as such in courts.

Indeed, since there is a collection of redistricting maps that can be considered valid or legal, it seems quite reasonable to attempt to output the most "typical" map. Inspired by social choice theory (in particular, the Kemeny rule [53, 127]), we propose a redistricting procedure in which the most typical (central) map among a given collection is selected. More precisely, we introduce a family of distance functions over redistricting maps and then select the map which minimizes the sum of distances to the other maps. In this chapter we include the proofs when they require arguments which are somewhat involved ([35] includes all of the proofs).

4.1 Related Work

Less than a decade ago, several early works ushered in the current era of Markov Chain Monte Carlo (MCMC) sampling techniques for gerrymandering detection [33, 128, 129]. Followup work has both refined these techniques and further analyzed their ability to approximate the target distribution. Authors of these works have been involved in court cases in Pennsylvania [120] and North Carolina [121] with sampling approaches being used to demonstrate that existing maps were outliers as evidence of partisan gerrymandering. One of the most recent works in this area introduces the **ReCom** tool [32] which was used by the Wisconsin People's Maps Commission and the Michigan Independent Citizens Redistricting Commission in the current redistricting cycle following the 2020 U.S. census [123]. Overall, these techniques have primarily been used to analyze and sometimes reject existing maps rather than draw new maps. However, we may think of them as at least narrowing the search space of maps drawn by legislatures. Along these lines, it has been shown that even the regulation of gerrymandering via outlier detection is subject to strategic manipulation [130].

On the automated redistricting side, many map drawing algorithms have favored optimization approaches and in particular, optimizing some notion of compactness while avoiding explicit use of partisan information. Approaches emphasizing compactness include balanced power diagrams [124], a k-median-based objective [131], and minimizing the number of cut edges in a planar graph [125]. Some works include partisan information for the sake of creating competitive districts (districts with narrow margins between the two main parties). The PEAR tool [126] balances nonpartisan criteria like compactness (defined by Polsby-Popper score [132]) with other criteria such as competitiveness and uses an evolutionary algorithm with some similarity to the random walks taken by MCMC sampling approaches. Other works go even further in the explicitly partisan direction. For example, [133] devises a game theoretic approach which aims to provide a map that is fair to the two dominant parties. Finally, there are methods which prefer simplicity such as the Splitline [134] algorithm which iteratively splits a state until the desired number of districts is reached.

In all of these approaches the aim is to automate redistricting, but it is difficult to determine whether the choices made are the "right" or "fairest" decisions. The question of whether optimizing properties such as compactness while ignoring partisan factors could result in partisan bias has been a concern for some time. [135] notes a comment by Justice Scalia suggesting that such a process could be biased against Democratic voters clustered in cities in Vieth v. Jubelirer [136]. For those that do take partisan bias into account, there are questions of whether purposely drawing competitive districts or giving a fair allocation to two parties are really beneficial to voters.

Finally, like [137] we are introducing a distance measure over redistricting maps. However, our distance is easy to compute and does not require solving a linear program. Further, our focus is on the implications of having a distance measure, i.e. the medoid and centroid maps that will be

introduced. Moreover, unlike [137] we can detect gerrymandered maps rigorously by specifying where they lie on a distance histogram without using an embedding method and using at least $50,000$ samples instead of only 100.

4.2 Problem Setup

A given state is modelled by a graph $G = (V, E)$ where each vertex $v \in V$ represents a voting block (*unit*). Each unit v has a weight $w(v) > 0$ which represents its population. Further, $\forall u, v \in V$ there is an edge $e = (u, v) \in E$ if and only if the two vertices are *connected* (geographically this means that units u and v share a boundary). The number of units is $|V| = n$. A *redistricting* (*redistricting map* or simply *map*) M is a partition of V into $k > 0$ many districts, i.e., $V = V_1 \cup V_2 \cdots \cup V_k$ where each V_i represents a district and $\forall i \in [k], |V_i| \neq 0$ and $\forall i, j \in [k], V_i \cap V_j = \emptyset$ if $i \neq j$. The redistricting map M is decided by the induced partition, i.e., $M = \{V_1, \ldots, V_k\}$. For a redistricting M to be considered valid, it must satisfy a collection of properties, some of which are specific to the given state. We use the most common properties as stated in [32, 125]: **(1)** *Compactness*: The given partitioning should have "compact" districts. Although there is no definitive mathematical criterion which decides compactness for districts, some have used common definitions such as Polsby-Popper or Reock Score [138]. Others have used a clustering criterion like the k-median objective [124] or considered the total number of cuts (number of edges between vertices in different districts) [32]. **(2)** *Equal Population*: To satisfy the desideratum of "one person one vote" each district should have approximately the same number of individuals. I.e., a given district V_i should satisfy $\sum_{v \in V_i} w(v) \in [(1-\epsilon)\frac{\sum_{v \in V} w(v)}{k}, (1+\epsilon)\frac{\sum_{v \in V} w(v)}{k}]$ where ϵ is a non-negative parameter relaxing the equal population constraint. **(3)**

Contiguity: Each district (partition) V_i should be a connected component, i.e., $\forall i \in [k]$ and $\forall u, v \in V_i$, v should be reachable from u through vertices which only belong to V_i.

Our proofs do not rely on these properties and therefore can likely even accommodate further desired properties.

Let $\mathcal{M}$ be the set of all valid maps. Let $\mathcal{D}(\mathcal{M})$ be a distribution over these maps. Furthermore, define a distance function over the maps $d : \mathcal{M} \times \mathcal{M} \to [0, \infty)$. Then the *population medoid map* is M^* which is a solution to the following:

$$M^* = \arg\min_{M \in \mathcal{M}} \mathop{\mathbb{E}}_{M' \sim \mathcal{D}(\mathcal{M})} [d(M, M')] \tag{4.1}$$

In words, the population medoid map is a valid map minimizing the expected sum of distances away from all valid maps according to the distribution $\mathcal{D}(\mathcal{M})$. This serves as a natural way to define a central or most typical map with respect to a given distance metric of interest.

Since we clearly operate over a sample (a finite collection) from $\mathcal{D}(\mathcal{M})$; therefore, we assume that the following condition holds:

Condition 4.2.1. *We can sample maps from the distribution $\mathcal{D}(\mathcal{M})$ in an independent and identically distributed (**iid**) manner in polynomial time.*

We note that although independence certainly does not hold over the sampling methods of [32, 33] since they use MCMC methods, it makes the derivations significantly more tractable. Further, the specific choice of the sampling technique is somewhat immaterial to our objective.

Based on the above condition, we can sample from the distribution $\mathcal{M}$ efficiently and obtain a finite set of maps $\mathcal{M}_T$ having T many maps, i.e., $|\mathcal{M}_T| = T$.

Now, we define the *sample medoid*, which is simply the extension of the population medoid,

but restricted to the given sample. This leads to the following definition:

$$\bar{M}^* = \arg\min_{M \in \mathcal{M}_T} \sum_{M' \in \mathcal{M}_T} d(M, M') \tag{4.2}$$

4.2.1 Distance over Redistricting Maps

Before we introduce a distance over maps, we note that a given map (partition) M can be represented using an "adjacency" matrix A in which $A(i,j) = 1$ if and only if $\exists V_\ell \in M : i, j \in V_\ell$ otherwise $A(i,j) = 0$. We note that this adjacency matrix can be seen as drawing an edge between every two vertices i, j that are in the same district, i.e., where $A(i,j) = 1$. It is clear that we can refer to a map by the partition M or the induced adjacency matrix A. Accordingly, we refer to the population medoid as M^* or A^* and the sample medoid as $\bar{M}^*$ or $\bar{A}^*$

We now introduce our distance family which is parametrized by a weight matrix Θ and have the following form:

$$d_\Theta(A_1, A_2) = \frac{1}{2} \sum_{i,j \in V} \theta(i,j) |A_1(i,j) - A_2(i,j)| \tag{4.3}$$

where we only require that $\theta(i,j) > 0, \forall i, j \in V$ where $\theta(i,j)$ is the (i,j) entry of Θ. For the simple case where $\theta(i,j) = 1, \forall i, j \in V$, our distance $d_{\mathbf{1}}(A_1, A_2)$ is equivalent to a Hamming distance over adjacency matrices. When $\theta(i,j) = 1, \forall i, j \in V$, we refer to the metric as the *unweighted distance*. We note that such a distance measure was used in previous work that considered adversarial attacks on clustering [139, 140].

Another choice of Θ that leads to a meaningful metric is the *population-weighted distance*

where $\theta(i,j) = w(i)w(j)$. This leads to $d_W(A_1, A_2) = \frac{1}{2}\sum_{i,j\in V} w(i)w(j)|A_1(i,j) - A_2(i,j)|$. The population-weighted distance takes into account the number of individuals being separated from one another when vertices i and j are separated from one another[2] by assigning a cost of $w(i)w(j)$. By contrast, the unweighted distance assigns the same cost regardless of the population values and thus has a uniform weight over the separation of units immaterial of the populations which they include.

Another choice of metric which is meaningful, could be of the form $\theta(i,j) = f(l(i,j))$ where $l(i,j)$ is the length of a shortest path between i and j and $f(.)$ is a positive decreasing function such as $f(l(i,j)) = e^{-l(i,j)}$. Such a metric would place a smaller penalty for separating vertices that are far away from each other.

In general, our distance has an edit distance interpretation. Specifically, if we were to draw edges between vertices according to the entries with 1 in the adjacency matrix, then given A_1 and A_2, the distance $d_\Theta(A_1, A_2)$ simply equals the minimum total weight (according to $\theta(i,j)$) of the edges that must be added and deleted to obtain A_2 from A_1. In the case of the unweighted distance, it is precisely equal to the minimum number of edges that have to be deleted from and added to A_1 to obtain A_2. See Figure 4.1 for an illustration.

4.3 Justification for Choosing a Central Map

Connection to the Kemeny Rule: We note that the Kemeny rule [53, 127] is the main inspiration behind our proposed framework. More specifically, if we have a set of alternatives and each individual votes by ranking the alternatives, then the Kemeny rule gives a method for aggregating

[2]Recall that each vertex (unit) is a voting block (AKA voter tabulation district) and units may contain different numbers of voters.

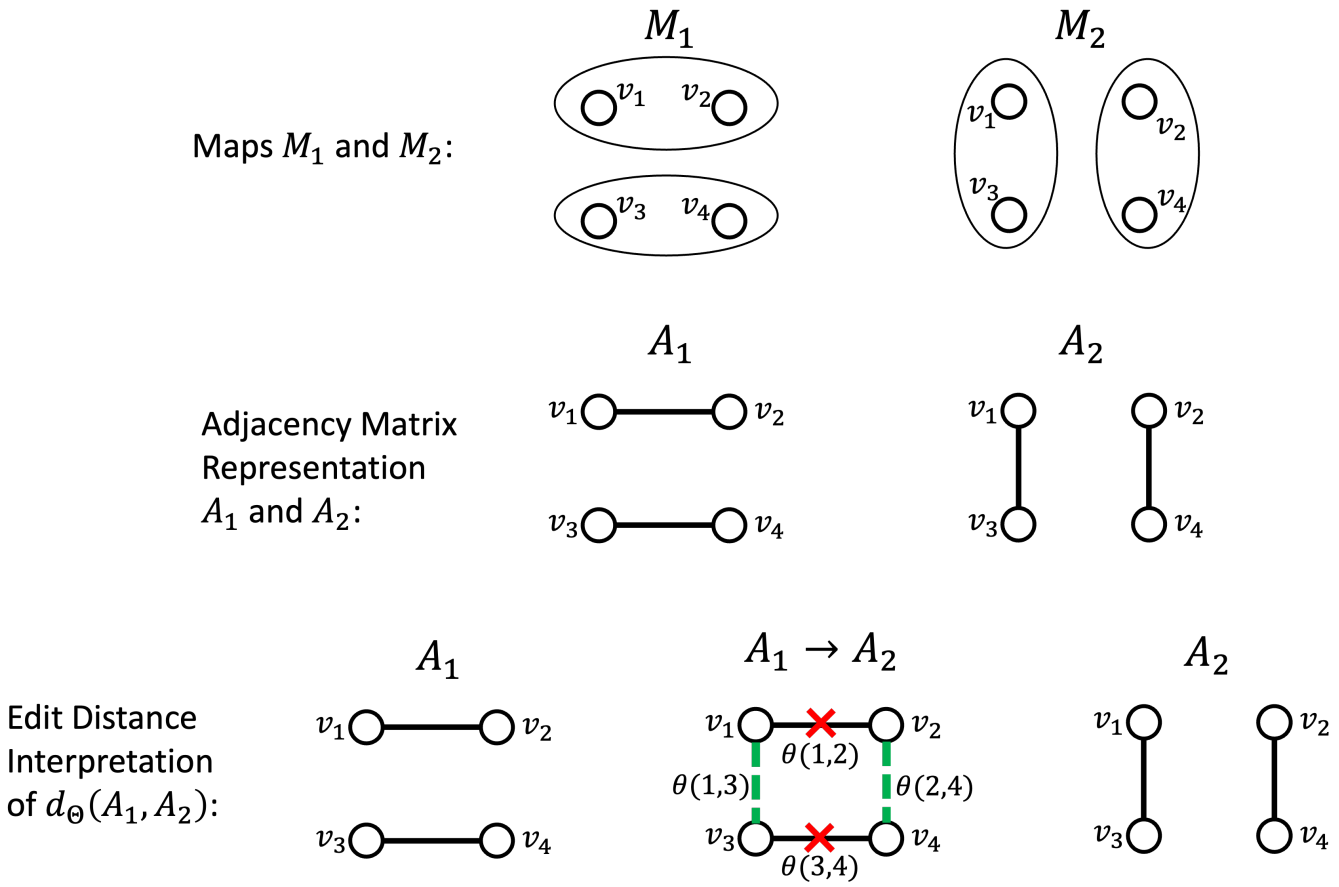


Figure 4.1: We are given a hypothetical state consisting of 4 vertices $V = \{v_1, v_2, v_3, v_4\}$ with M_1 and M_2 being two valid redistricting maps. The adjacency matrices A_1,A_2, and edit distance interpretation of $d_\Theta(A_1, A_2)$ are demonstrated. Note that $d_\Theta(A_1, A_2) = \theta(1,2) + \theta(3,4) + \theta(1,3) + \theta(2,4)$ which is exactly the minimum sum of edge weights that need to be deleted and added to obtain A_2 from A_1.

the resulting collection of rankings. This is done by introducing a distance measure over rankings (the Kendall tau distance [141]) and then choosing the ranking which minimizes the sum of distances away from the other rankings in the collection as the aggregate ranking.

Although we do not deal with rankings here, we follow a similar approach to the Kemeny rule as we introduce a distance measure over redistricting maps and choose the map which minimizes the sum of the distances as the aggregate map. In fact, recently there has been significant citizen engagement in drawing redistricting maps. For example, in the state of Maryland an

executive order from the governor has established a web page to collect citizen submissions of redistricting maps [142]. If each member of a committee was to vote for exactly one map in the given submitted maps, then if we interpret the probability $p_{M'}$ for a map $M' \in \mathcal{M}$ to be the number of votes it received from the total set of votes, then the medoid map M^* (similar to the Kemeny ranking) would be the map which minimizes the weighted sum of distances from the set of maps voted on. We include this result as a proposition and its proof follows directly from the definition we gave above:

Proposition 4.3.1. *Suppose we have a committee of $\mathcal{T}$ many voters and that each voter votes for one map from a subset of all possible valid maps $\mathcal{M}$, then given a map M', if we assign it a probability $p_{M'} = \frac{\sum_{\tau=1}^{\mathcal{T}} v_{\tau,M'}}{\mathcal{T}}$ where $v_{\tau,M'} \in \{0,1\}$ is the vote of member τ for map M', then the medoid map $M^* = \arg\min_{M \in \mathcal{M}} \mathbb{E}_{M' \sim p_{M'}}[d(M, M')]$ is the map that minimizes the sum of distances from the set of valid maps where the distance to each map is weighted by the total votes it receives.*

Connection to Distance and Clustering Based Outlier Detection: The medoid map by virtue of minimizing the sum of distances can be considered a central map. Accordingly, one may consider using the medoid map to test for gerrymandering in a manner similar to distance and clustering based outlier detection [143, 144]. More specifically, given a large ensemble of maps, if the enacted map is faraway from the medoid[3] in comparison to the ensemble then this suggests possible gerrymandering. In fact, we carry experiments on the states of North Carolina and Pennsylvania (both of which have had enacted maps which were considered gerrymandered) and we indeed find the gerrymandered maps to be faraway whereas the remedial maps are much

[3] In our experiments, we actually use the centroid instead of the medoid map.

closer in terms of distance.

4.4 Algorithms and Theoretical Guarantees

We show our linear time algorithm for obtaining the sample medoid in subsection 4.4.1. In 4.4.2 we define the population centroid, derive sample complexity guarantees for obtaining it, and show that its (i, j) entry equals the probability of having i and j in the same district. Finally, in 4.4.3 we discuss obtaining the population medoid. Specifically, we show that even for a simple one dimensional distribution an arbitrarily large sample is not sufficient for obtaining the population medoid.

First, before we introduce our algorithms we show that our distance family is indeed a metric, i.e. satisfies the properties of a metric:

Proposition 4.4.1. *For all Θ such that $\forall i, j, \theta(i, j) > 0$, the following distance function is a metric.*

$$d_{\Theta}(A_1, A_2) = \frac{1}{2} \sum_{i,j \in V} \theta(i, j) |A_1(i, j) - A_2(i, j)|$$

4.4.1 Obtaining the Sample Medoid

We note that in general obtaining the sample medoid is not scalable since it usually takes quadratic time [145] in the number of samples, i.e. $\Omega(T^2)$. An $O(T^2)$ run time can be easily obtained through a brute-force algorithm which for every map calculates the sum of the distances from other maps and then selects the map with the minimum sum. However, for our family of distances $d_{\Theta}(.,.)$ we show that the medoid map is the closest map to the centroid map and show a simple algorithm that runs in $O(T)$ time for obtaining the sample medoid. The fundamental

cause behind this speed up is an equivalence between the Hamming distance over binary vectors and the square of the Euclidean distance which is still maintained with our generalized distance. Before introducing the theorem we define $d_{2,\Theta}(A_1, A_2) = \frac{1}{2}\sum_{i,j\in V}\theta(i,j)(A_1(i,j) - A_2(i,j))^2$ where the absolute has been replaced by a square. Now we state the decomposition theorem:

Theorem 4.4.2. *Given a collection of redistricting maps $A_1, \ldots, A_T$, the sum of distances of the maps from a fixed redistricting map A' equals the following:*

$$\sum_{t=1}^{T} d_\Theta(A_t, A') = \sum_{t=1}^{T} d_{2,\Theta}(A_t, \bar{A}_c) + T\, d_{2,\Theta}(\bar{A}_c, A') \tag{4.4}$$

where $\bar{A}_c = \frac{1}{T}\sum_{t=1} A_t$.

Proof. We begin with the following lemma:

Lemma 4.4.3. *For any A_1, A_2 that are binary matrices (entries either 0 or 1), with $d_{2,\Theta}(A_1, A_2) = \frac{1}{2}\sum_{i,j\in V}\theta(i,j)(A_1(i,j) - A_2(i,j))^2$, then we have that $d_\Theta(A_1, A_2) = d_{2,\Theta}(A_1, A_2)$.*

Proof. The proof is immediate since A_1 and A_2 are binary. □

It may seem redundant to introduce a new definition $d_{2,\Theta}(.,.)$ since by Lemma 4.4.3, they are equivalent. However, we will shortly be using $d_{2,\Theta}(.,.)$ over matrices which are not necessarily binary, clearly then we might have $d_\Theta(A_1, A_2) \neq d_{2,\Theta}(A_1, A_2)$.

We then introduce next lemma:

Lemma 4.4.4. *For any two matrices (not necessarily binary), the following holds:*

$$d_{2,\Theta}(A_1, A_2) = \frac{1}{2}\left\|A_1^\Theta - A_2^\Theta\right\|_2^2 \tag{4.5}$$

where $A_s^{\Theta}(i,j) = \sqrt{\theta(i,j)}A_s(i,j), \forall s \in \{1,2\}$ *and* $\left\|A_1^{\Theta} - A_2^{\Theta}\right\|_2^2$ *is the square of the* ℓ_2*-norm of the vectorized form of the matrix* $(A_1^{\Theta} - A_2^{\Theta})$.

Proof.

$$\begin{aligned} d_{2,\Theta}(A_1, A_2) &= \frac{1}{2}\sum_{i,j\in V}\theta(i,j)(A_1(i,j) - A_2(i,j))^2 \\ &= \frac{1}{2}\sum_{i,j\in V}(\sqrt{\theta(i,j)}A_1(i,j) - \sqrt{\theta(i,j)}A_2(i,j))^2 \\ &= \frac{1}{2}\left\|A_1^{\Theta} - A_2^{\Theta}\right\|_2^2 \end{aligned}$$

□

With the lemmas above, we can now prove the decomposition theorem:

$$\begin{aligned} \sum_{t=1}^{T} d_{\Theta}(A_t, A') &= \sum_{t=1}^{T} d_{2,\Theta}(A_t, A') \qquad \text{(using Lemma 4.4.3)} \\ &= \sum_{t=1}^{T}\frac{1}{2}\left\|A_t^{\Theta} - A'^{\Theta}\right\|_2^2 \qquad \text{(using Lemma 4.4.4)} \\ &= \frac{1}{2}\sum_{t=1}^{T}\left\|A_t^{\Theta} - \bar{A}_c^{\Theta} + \bar{A}_c^{\Theta} - A'^{\Theta}\right\|_2^2 \\ &= \frac{1}{2}\sum_{t=1}^{T}\left[(A_t^{\Theta} - \bar{A}_c^{\Theta} + \bar{A}_c^{\Theta} - A'^{\Theta})^{\intercal}(A_t^{\Theta} - \bar{A}_c^{\Theta} + \bar{A}_c^{\Theta} - A'^{\Theta})\right] \\ &= \sum_{t=1}^{T}\frac{1}{2}\left[\left\|A_t^{\Theta} - \bar{A}_c^{\Theta}\right\|_2^2 + \left\|\bar{A}_c^{\Theta} - A'^{\Theta}\right\|_2^2\right] + \left(\sum_{t=1}^{T}(A_t^{\Theta} - \bar{A}_c^{\Theta})\right)^{\intercal}(\bar{A}_c^{\Theta} - A'^{\Theta}) \\ &= \sum_{t=1}^{T}\frac{1}{2}\left\|A_t^{\Theta} - \bar{A}_c^{\Theta}\right\|_2^2 + \frac{T}{2}\left\|\bar{A}_c^{\Theta} - A'^{\Theta}\right\|_2^2 + (T\,\bar{A}_c^{\Theta} - T\,\bar{A}_c^{\Theta})^{\intercal}(\bar{A}_c^{\Theta} - A'^{\Theta}) \\ &= \sum_{t=1}^{T} d_{2,\Theta}(A_t, \bar{A}_c) + T\,d_{2,\Theta}(\bar{A}_c, A') \end{aligned}$$

Note that in the fourth line we take the dot product with the matrices being in vectorized form

Algorithm 15 Finding the Sample Medoid

Input: $\mathcal{M}_T = \{A_1, \ldots, A_T\}$, $\Theta = \{\theta(i,j) > 0, \forall i, j \in V\}$.
1: Calculate the centroid map $\bar{A}_c = \frac{1}{T}\sum_{t=1}^{T} A_t$.
2: Pick the map $\bar{A}^* \in \mathcal{M}_T$ which minimizes the $d_{2,\Theta}$ distance from the centroid $\bar{A}_c$, i.e. $\bar{A}^* = \arg\min_{A \in \mathcal{M}_T} d_{2,\Theta}(A, \bar{A}_c)$.
return $\bar{A}^*$

and that $\bar{A}_c^{\Theta} = \frac{1}{T}\sum_{t=1}^{T} A_t^{\Theta}$. Note that it follows that $\bar{A}_c^{\Theta}(i,j) = \sqrt{\theta(i,j)}\,\bar{A}_c(i,j)$. □

Notice that the above theorem introduces the centroid map $\bar{A}_c$ which is simply equal to the empirical mean of the adjacency maps. It should be clear that with the exception of trivial cases the centroid map $\bar{A}_c$ is not a valid adjacency matrix, since despite being symmetric it would have fractional entries between 0 and 1. Hence, the centroid map also does not lead to a valid partition or districting. Moreover, we note that it is more accurate to call $\bar{A}_c$ the sample centroid, as opposed to the population centroid A_c (see subsection 4.4.2) which we would obtain with an infinite number of samples.

The above theorem leads to Algorithm 15 with the following remark:

Remark 4.4.5. *Algorithm 1 returns the correct sample medoid and runs in $O(T)$ time.*

We note that calculating the sample medoid in algorithm 1 has no dependence on the generating method. Therefore, if a set of maps are produced through any mechanism and are considered to be representative and sufficiently diverse, then algorithm 1 can be used to obtain the sample medoid in time that is linear in the number of samples.

4.4.2 Sample Complexity for Obtaining the Population Centroid

In the previous section we introduced the sample centroid $\bar{A}_c$ which is simply equal to the empirical mean that we get by taking the average of the adjacency matrices, i.e. $\bar{A}_c = \frac{1}{T}\sum_{t=1}^{T} A_t$.

We now consider the population centroid $A_c = \lim_{T\to\infty} \sum_{t=1}^{T} A_t$. Clearly, by the law of large numbers [146], we have $A_c(i,j) = \mathbb{E}[A(i,j)]$. It is also clear that A_c has an interesting property, specifically the (i,j)-entry equals the probability that i and j are in the same district:

Proposition 4.4.6. $A_c(i,j) = \Pr[i \textit{ and } j \textit{ in the same district}]$.

Now we show that with a sufficient number of samples, the sample centroid converges to the population centroid entry-wise and in terms of the $d_{2,\Theta}$ value. Specifically, we have the following proposition:

Proposition 4.4.7. *If we sample $T \geq \frac{1}{\epsilon^2} \ln \frac{n}{\delta}$ **iid** samples, then with probability at least $1-\delta$, we have that $\forall i,j \in V : |\bar{A}_c(i,j) - A_c(i,j)| \leq \epsilon$. Further, let $\kappa = \max_{i,j\in V} \sqrt{\theta(i,j)}$, if we have $T \geq \frac{\kappa n^2}{\epsilon} \ln \frac{n}{\delta}$ **iid** samples, then $d_{2,\Theta}(\bar{A}_c, A_c) \leq \epsilon$ with probability at least $1-\delta$.*

Proof. For a given $i,j \in V$ by the Hoeffding bound we have that:

$$\begin{aligned}
&\Pr[|\bar{A}_c(i,j) - A_c(i,j)| \leq \epsilon] \geq 1 - 2e^{-2\epsilon^2 T} \\
&\geq 1 - 2e^{-2\epsilon^2 \frac{1}{\epsilon^2} \ln \frac{n}{\delta}} \geq 1 - 2(e^{2\ln \frac{n}{\delta}})^{-1} \geq 1 - 2\frac{\delta^2}{n^2}
\end{aligned}$$

Now we calculate the following event:

$$\begin{aligned}
&\Pr(\{\forall i,j \in V : |\bar{A}_c(i,j) - A_c(i,j)| \leq \epsilon\}) \\
&= 1 - \Pr(\{\exists i,j \in V : |\bar{A}_c(i,j) - A_c(i,j)| > \epsilon\}) \\
&\geq 1 - \sum_{i,j\in V} 2\frac{\delta^2}{n^2} \geq 1 - 2\delta^2 \frac{\left(\frac{n^2-n}{2}\right)}{n^2} \geq 1 - \delta^2 \geq 1 - \delta \qquad (\text{ since } \delta \in (0,1))
\end{aligned}$$

Now we prove the second part. By applying the previous result with ϵ set to $\frac{\sqrt{\epsilon}}{\sqrt{\rho}n}$, we get that with

probability at least $1-\delta$, $|\bar{A}_c(i,j) - A_c(i,j)| \leq \frac{\sqrt{\epsilon}}{\sqrt{\rho}n^2}$. It follows that:

$$\begin{aligned} d_{2,\Theta}(\bar{A}_c, A_c) &= \frac{1}{2}\sum_{i,j\in V} \theta(i,j)(\bar{A}_c(i,j) - A_c(i,j))^2 \\ &\leq \frac{1}{2}\sum_{i,j\in V} \theta(i,j)\Big(\frac{\sqrt{\epsilon}}{\sqrt{\rho}n}\Big)^2 \leq \frac{1}{2}\sum_{i,j\in V}\frac{\epsilon}{n^2} \\ &\leq \frac{1}{2}\frac{\epsilon}{n^2}\frac{n^2-n}{2} \leq \epsilon \end{aligned}$$

□

4.4.3 Obtaining the Population Medoid

Having found the sample centroid $\bar{A}_c$ and shown that it is a good estimate of the population centroid A_c, we now show that we can obtain a good estimate of the population medoid by solving an optimization problem. Specifically, assuming that we have the population centroid A_c, then the population medoid is simply a valid redistricting map A which has a minimum $d_{2,\Theta}(A, A_c)$ value. This follows immediately from Theorem 4.4.2. More interestingly, we show in fact that this optimization problem is a constrained instance of the min k-cut problem:

Theorem 4.4.8. *Given the population centroid A_c, the population medoid A^* can be obtained by solving a constrained min k-cut problem.*

Proof. From Theorem 4.4.2, the population medoid is a valid redistricting map A for which $d_{2,\Theta}(A, A_c)$ is minimized. Note that since A is a redistricting map, unlike A_c it must be a binary matrix. Therefore, $|A(i,j) - A_c(i,j)| = (1 - A_c(i,j)) + (2\,A_c(i,j) - 1)(1 - A(i,j))$, where this identity can be verified by plugging 0 or 1 for $A(i,j)$ and seeing that it leads to an equality. Define the matrix B as a "complement" of A. Specifically, $B(i,j) = 1 - A(i,j)$. It follows that

$B(i,j) = 1$ if and only if i and j are in different partitions and $B(i,j) = 0$ otherwise. Clearly, B is a binary matrix. We can obtain the following:

$$
\begin{aligned}
d_{2,\Theta}(A, A_c) &= \frac{1}{2} \sum_{i,j \in V} \theta(i,j)(A(i,j) - A_c(i,j))^2 \\
&= \frac{1}{2} \sum_{i,j \in V} \theta(i,j)\big((1 - A_c(i,j)) + (2\,A_c(i,j) - 1)(1 - A(i,j))\big)^2 \\
&= \frac{1}{2} \sum_{i,j \in V} \theta(i,j)\big((1 - A_c(i,j)) + (2\,A_c(i,j) - 1)B(i,j)\big)^2 \\
&= \frac{1}{2} \sum_{i,j \in V} \theta(i,j)\big((1 - A_c(i,j))^2 + 2(1 - A_c(i,j))(2\,A_c(i,j) - 1)B(i,j) + (2\,A_c(i,j) - 1)^2 B^2(i,j)\big) \\
&= \frac{1}{2} \sum_{i,j \in V} \theta(i,j)\big((1 - A_c(i,j))^2 + 2(1 - A_c(i,j))(2\,A_c(i,j) - 1)B(i,j) + (2\,A_c(i,j) - 1)^2 B(i,j)\big) \\
&= \Big[\frac{1}{2} \sum_{i,j \in V} \theta(i,j)({A_c}^2(i,j) - 2\,A_c(i,j) + 1)\Big] - \Big[\frac{1}{2} \sum_{i,j \in V} \theta(i,j)\big(1 - 2\,A_c(i,j)\big)B(i,j)\Big]
\end{aligned}
$$

Note that the first sum in the last equality is a constant and has no dependence on B. Hence to minimize $d_{2,\Theta}(A, A_c)$, we maximize the following:

$$\max_{B} \sum_{i,j \in V} s(i,j)B(i,j) \tag{4.6}$$

$$\text{s.t. } B \text{ is a } k \text{ partition that leads to a valid redistricting map} \tag{4.7}$$

where the weight $s(i,j)$ is equal to $s(i,j) = \frac{1}{2}\theta(i,j)\big(1 - 2\,A_c(i,j)\big)$. Clearly, this is a constrained max k-cut instance where the partition has to be a valid redistricting map. □

If we have a good estimate $\bar{A}_c$ of the population centroid A_c, then we can solve the above optimization using $\bar{A}_c$ instead of A_c and obtain an estimate of the population medoid $\bar{A}^*$ instead

of the true population medoid A^* and bound the error of that estimate. The issue is that the min k-cut problem is NP-hard [147, 148] [4]. Further, the existing approximation algorithms assume non-negativity of the weights. Even if these approximation algorithms can be tailored to this setting, the additional constraints on the partition being a valid redistricting (each partition being contiguous, of equal population, and compact) make it quite difficult to approximate the objective. In fact, excluding the objective and focusing on the constraint alone, only the work of [125] has produced approximation algorithm for redistricting maps but has done that for the restricted case of grid graphs. Further, while there exists heuristics for solving min k-cut for redistricting maps they only scale to at most around 500 vertices [149].

Having shown the difficulty in obtaining the population medoid by solving an optimization problem, it is reasonable to wonder whether we can gain any guarantees about the population medoid by sampling. We show a negative result. Specifically, the theorem below shows that we cannot guarantee that we can estimate the sample medoid of a distribution with high probability by choosing a sampled map even if we sample an arbitrarily large number of maps. This implies as a corollary that the sample medoid does not converge to the population medoid in contrast to the centroid (see Proposition 4.4.7).

Theorem 4.4.9. *For any arbitrary T many **iid** samples $\{A_1, \ldots, A_T\}$ there exists a distribution over a set of redistricting maps such that: (1) $\Pr[\min_{A \in \{A_1,\ldots,A_T\}} d(A, A^*) \geq 0.331] \geq \frac{2}{3}$ and (2) $\Pr[\min_{A \in \{A_1,\ldots,A_T\}} f(A) \geq 1.1 f(A^*)] \geq \frac{2}{3}$ where $f(.)$ is the medoid cost function.*

Proof. Consider the hypothetical state shown in Figure 4.2 where vertices v_1 and v_4 are further

[4] Note that in the case of the min k-cut problem of Eq (4.6) the edge weights $s(i,j) = \frac{1}{2}\theta(i,j)\big(1 - 2\,A_c(i,j)\big)$ can be negative while the min k-cut problem is generally stated with non-negative weights. Nevertheless, Eq (4.6) still minimizes a cut objective and the non-negative weight min k-cut instance is trivially reducible to a min k-cut instance with negative and non-negative weights.

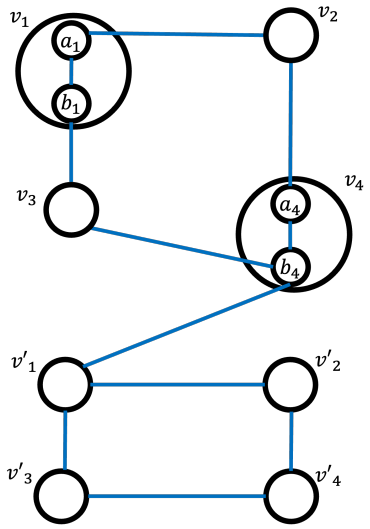


Figure 4.2: The graph shows a hypothetical state. Blue edges indicate that the vertices are adjacent geographically. All vertices have a weight (population) of 1, except for states $\{a_1, b_1, a_4, b_4\}$ which have a weight of $\frac{1}{2}$.

subdividing into two vertices each. We wish to divide the state into 2 districts ($k = 2$). Since each vertex has a weight of 1, except vertices $\{a_1, b_1, a_4, b_4\}$ which each have a weight of $\frac{1}{2}$, then each district should have a population of 2 to enforce the equal population rule with a tolerance less than 0.25.

Denoting the set of all vertices by V and letting $V' = \{a_1, b_1, a_4, b_4\}$, then the weight parameters of our weighted distance measure are defined as follows:

$$\theta(i,j) = \begin{cases} \epsilon & \text{if } i \,\&\, j \in V' \\ \frac{1}{2} & \text{if } i \in V - V', j \in V' \text{ or } i \in V', j \in V - V' \\ 1 & \text{otherwise} \end{cases}$$

Where $0 < \epsilon \leq 1$. Now, consider the maps M_1, M_2, M_3, and M_4 shown in Figure 4.3. Based on the definition of the weighted distance measure, it is not difficult to see that given maps

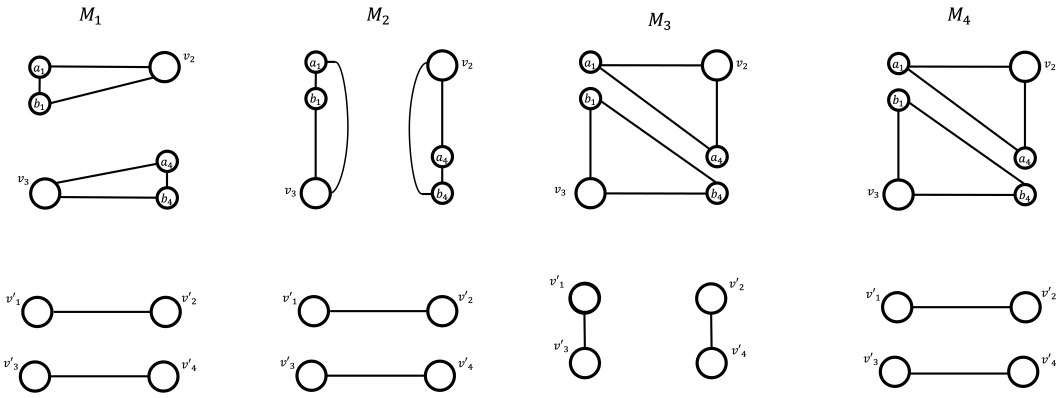


Figure 4.3: Maps M_1, M_2, M_3, and M_4. Vertices in the same district are connected with edges.

M_s and M_t, then $d_\Theta(M_s, M_t)$ can be computed visually by drawing the adjacent graphs of M_s and M_t and then finding the minimum number of edges that have to be deleted and added to M_s to produce M_t and adding the weighted $\theta(i.j)$ of these edges. By following this procedure, we can show that $d_\Theta(M_1, M_3) = 6 + 4\epsilon$ for example as shown in Figure 4.4. Here we list all distances:

$$d_\Theta(M_1, M_2) = 4$$

$$d_\Theta(M_1, M_4) = d_\Theta(M_2, M_4) = 2 + 4\epsilon$$

$$d_\Theta(M_1, M_3) = d_\Theta(M_2, M_4) = (2 + 4\epsilon) + 4 = 6 + 4\epsilon$$

$$d_\Theta(M_3, M_4) = 4$$

Given a map M, the medoid cost function is defined as $f(M) = \sum_{M' \in \mathcal{M}} p_{M'} d(M, M')$. Let the probabilities for the redistricting maps be assigned as follows: $p_1 = p_2 = p_3 = \frac{1-\delta}{3}$ whereas $\Pr[\text{Sampling a Map } M \notin \{M_1, M_2, M_3\}] = \delta > 0$. Accordingly, $p_5 \leq \delta$.

Further, since $\theta(i, j) \leq 1$, $\forall i, j \in V$, then maximum distance between any redistricting

maps D can be upper bounded by the highest number of edges that can be deleted and added from one map to produce another, therefore $D \leq 2\binom{|V|}{2} = |V|(|V|-1) = 10 \times 9 = 90$ since $|V| = 10$ (see Figure 4.2). The medoid cost function can be lower bounded for M_1, M_2, and M_3 and upper bounded for M_4 as shown below:

$$
\begin{aligned}
f(M_1) &> \frac{1-\delta}{3}[d(M_1, M_2) + d(M_1, M_3)] = \frac{1-\delta}{3}(10 + 4\epsilon) \\
f(M_2) &> \frac{1-\delta}{3}[d(M_1, M_2) + d(M_2, M_3)] = \frac{1-\delta}{3}(10 + 4\epsilon) \\
f(M_3) &> \frac{1-\delta}{3}[d(M_1, M_3) + d(M_2, M_3)] = \frac{1-\delta}{3}(12 + 8\epsilon) \\
f(M_4) &< \frac{1-\delta}{3}[d(M_1, M_4) + d(M_2, M_4) + d(M_3, M_4)] + \delta D = \frac{1-\delta}{3}(8 + 8\epsilon) + 90\delta \leq \frac{1-\delta}{3}(9 + 8\epsilon)
\end{aligned}
$$

Where the last inequality was obtained by setting $\delta < \frac{1}{271}$ since it follows that $90\delta < \frac{1-\delta}{3}$. From the above bounds it follows that the population medoid cannot cannot be M_1, M_2, or M_3.

Set $\epsilon = \frac{1}{1000}$ and $\delta \leq \frac{1}{1000} < \frac{1}{271}$ and with $1 - (p_1 + p_2 + p_3) = \delta$, then $\frac{1-\delta}{3}(1 - 4\epsilon) > 0.331$ and $\frac{(10+4\epsilon)}{(9+8\epsilon)} > 1.11$. With the population medoid denoted by M^*, then we have: $\min_{i \in \{1,2,3\}} \frac{f(M_i)}{f(M^*)} \geq \min_{i \in \{1,2,3\}} \frac{f(M_i)}{f(M_4)} \geq \frac{\frac{1-\delta}{3}(10+4\epsilon)}{\frac{1-\delta}{3}(9+8\epsilon)} > 1.11$. Further, it follows by the triangle inequality that $\forall i \in \{1,2,3\} : f(M_i) \leq f(M^*) + d(M_i, M^*)$, thus $d(M_i, M^*) \geq \frac{1-\delta}{3}(1 - 4\epsilon)$, since otherwise $f(M^*)+d(M_i, M^*) \leq f(M_4)+d(M_i, M^*) < \frac{1-\delta}{3}(9+8\epsilon)+\frac{1-\delta}{3}(1-4\epsilon) = \frac{1-\delta}{3}(10+4\epsilon)$ which would be a contradiction.

From the above we have shown that, $\forall i \in \{1,2,3\} : d(M_i, M^*) \geq \frac{1-\delta}{3}(1 - 4\epsilon) > 0.331$ and $\min_{i \in \{1,2,3\}} \frac{f(M_i)}{f(M^*)} \geq 1.11$, therefore to prove parts (1) and (2) of the theorem it is sufficient to upper bound the probability of sampling a map that is not in $\{M_1, M_2, M_3\}$ in T **iid** samples by

$\frac{1}{3}$. This leads to the following:

$$\begin{aligned}
&\Pr[\text{Obtaining a map } M \notin \{M_1, M_2, M_3\} \text{ in a given } T \textbf{ iid} \text{ samples}] \\
&= 1 - \Pr[\text{No map } M \in \{M_1, M_2, M_3\} \text{ in the given } T \textbf{ iid} \text{ samples}] \\
&= 1 - (1-\delta)^T \leq \frac{1}{3}
\end{aligned}$$

Therefore, from the above we should have $\delta = \min\{\frac{1}{1000}, 1 - \sqrt[T]{\frac{2}{3}}\}$ to satisfy both parts of the theorem.

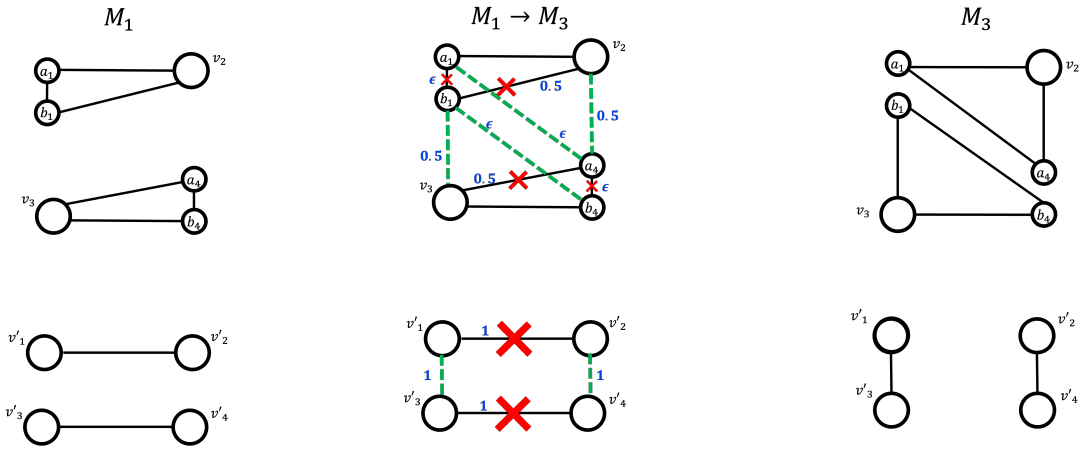


Figure 4.4: The first map is M_1 and the last is M_3. The middle map shows the edges the should be deleted from M_1 (marked with X) and the edges should be added to M_1 (dashed green edges) to produce M_3. The weight of each edge that is deleted or added is shown next to it in blue. By adding the weights we get that $d_\Theta(M_1, M_3) = 6 + 4\epsilon$.

□

We therefore, use a heuristic to find the medoid as mentioned in section 2.5.6.

Remark: In Theorem 4.4.9 the probability of "failure" is set to $\frac{2}{3}$ but this is arbitrary as we can make it arbitrarily large by choosing smaller values of δ. But our objective was simply to show that no sampled map would converge to the population medoid or would have a medoid

cost function value that converges to the value of medoid cost function of the population medoid. Further, the theorem would hold if the population medoid is sampled with probability zero, but in our proofs we allowed the population medoid to be sampled with non-zero probability to show that the negative result would still hold even if we were to assume that the population medoid is sampled with non-zero probability.

4.5 Experiments

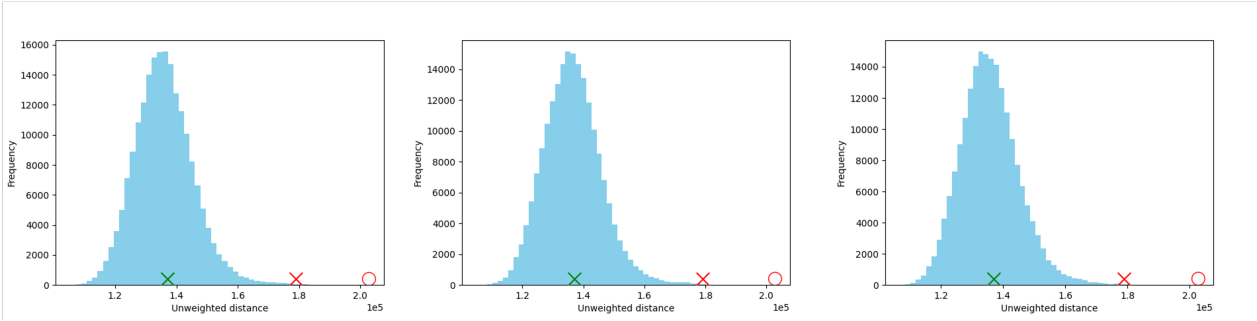


Figure 4.5: Distance histograms for NC using the unweighted distance measure. Different plots correspond to different seeds. For NC the distances of gerrymandered maps are indicated with red markers whereas the distances of the remedial maps are indicated with green markers (the ○ and the X are for 2011 and 2016 enacted maps, respectively).

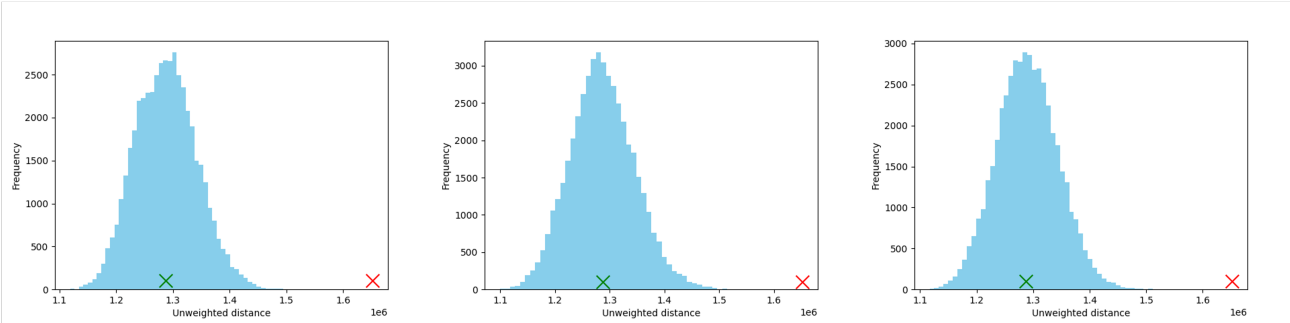


Figure 4.6: Distance histograms for PA, the distances of gerrymandered maps are indicated with red markers whereas the distances of the remedial maps are indicated with green markers.

We conduct our experiments over 3 states. Specifically, North Carolina (NC), Maryland (MD), and Pennsylvania's (PA). The number of voting units (vertices) and districts are around

2,700, 1,800, and 8,900 for NC, MD, and PA, respectively. Further, the number of districts are 13, 8, and 18 for NC, MD, and PA, respectively[5]. Accordingly, PA is the largest state whereas MD is the smallest. He focus on the results for NC here (see [35] for the details of the PA and MD results). We note that qualitatively all 3 states behave similarly. To generate a collection of maps, we use the Recombination algorithm **ReCom** from [32] whose implementation is available online. We note that **ReCom** is a Markov Chain Monte Carlo (MCMC) sampling method and hence the generated samples are not actually **iid**. While this means that Condition (4.2.1) does not hold, we believe that our theorems still have utility and that future work can address more realistic sampling conditions. Moreover, we always exclude the first 2,000 from any calculation as these are considered to be "burn-in" samples[6]. Throughout this section when we say distance we mean $d_{2,\Theta}(.,.)$ instead of $d_{\Theta}(.,.)$. Full experimental details are shown in [35].

Convergence of the Centroid: We note that previous work such as [32, 150] had used the **ReCom** algorithm for estimating statistics such as the histogram of election seats won by a party and has noted that using 50,000 samples is sufficient for accurate results. However, our setting is more challenging. Specifically, the centroid includes $\Omega(n)$ entries where n is the total number of voting units (vertices) whereas the election histogram includes only k entries where k is the number of districts and usually orders of magnitude smaller than the number of voting units. We sample 200,000[7] maps instead to estimate the centroid. Here we emphasize the importance of our linear-time algorithm since using a quadratic-time algorithm on samples of the order of even

[5]Note that for PA the number of districts has been reduced by one to 17 districts after the 2020 census. However, since we use past election results we have 18 districts.

[6]In MCMC, the chain is supposed to converge to a stationary distribution after some number of steps. These number of steps are called the mixing time. Although for **ReCom** the mixing time has not been theoretically calculated, empirically it seems that 2,000 steps are sufficient.

[7]A smaller number is used for PA and MD but we find that it is empirically sufficient.

50,000 could be computationally forbidding. Following similar practice to [121] for verifying convergence, we repeat the procedure (sampling using **ReCom** and estimating the centroid) for a total of three times for each state where we start from a different seed map each time and verify that all three runs result in essentially the same centroid estimate.

To verify the closeness of the different centroid estimates, we calculate the distances between them and compare them to their distances from sampled redistricting maps using **ReCom**. We find that the centroids are orders of magnitude closer to each other than to any other sampled map. For example, the maximum unweighted distance between any two centroids is less than 130 whereas the minimum unweighted distance between any of the three centroids and any sampled map is more than $100,000$ which is three orders of magnitude higher. Similarly, the maximum weighted distance between any two centroids is less than 1.6×10^9 whereas the mimimum weighted distance between a sampled amp and centroid is at least 1.3×10^{12} which is again three orders of magnitude higher.

Distance Histogram and Detecting Gerrymandered Maps: For each state we plot the distance histogram from its centroid. More specifically, having estimated the centroid $\bar{A}_c$, we sample 200,000 maps and calculate $d_{2,\Theta}(\bar{A}_c, A_t)$ where A_t is the t^{th} sampled map. Figures 4.5 and 4.6 shows the unweighted distance histogram for NC and PA, respectively. The histogram appears like a normal distribution peaking at the middle (around the mean) and falling almost symmetrically away from the middle. This also indicates that while the centroid minimizes the sum of $d_{2,\Theta}(\bar{A}_c, A_t)$ distances, the maps do not actually concentrate around it as otherwise the histogram would have had a peak in the beginning at small distance values. Interestingly, the histogram has a similar shape for both distances (unweighted and weighted). Further, this shape of the

histogram remains unchanged accross the different seeds.

Furthermore, previous work has used similar sampling methods to detect gerrymandered maps [33, 120, 121]. In essence these paper demonstrate that the election outcome achieved by the enacted map is rare to happen in comparison to the large sampled ensemble of redistricting maps. Using similar logic, we find that we can also detect gerrymandered maps. Specifically, the 2011 and 2016 enacted maps of NC were widely considered to be gerrymandered and we find both maps to be at the right tail of the histogram and very faraway from the centroid. In contrast, to a remedial NC map that was drawn by a set of retired judges [121] which is much closer to the centroid (see Figure 4.5 red and green marked points). Similarly for PA we find that the gerrymandered map of 2011 which was struck down by the supreme court [34] is also at the tail of the histogram whereas the remedial map has a much reasonable distance value. Quite interestingly, all gerrymandered maps are in the 99th percentile in terms of distance (for both distance measures and across 3 seeds).

This suggests that we indeed have a method for detecting gerrymandered maps which in comparison to previous methods has the advantage of not needing election results (only a reasonable distance measure) and is very interpretable. Further, it is reasonable to consider this as setting a new rule when drawing redistricting maps or at least a guideline: the drawn map should not be very far away from the centroid.

Finding the medoid: We discuss the results for the unweighted distance. Since we have shown in subsection 4.4.3 that the medoid cannot be obtained by sampling, we follow a heuristic that consists of these steps: (1) Sample $200{,}000$ maps and pick the one closest to the centroid A_{closest}. (2) Start the **ReCom** chain from A_{closest} but given a specific state (redistricting map) we only

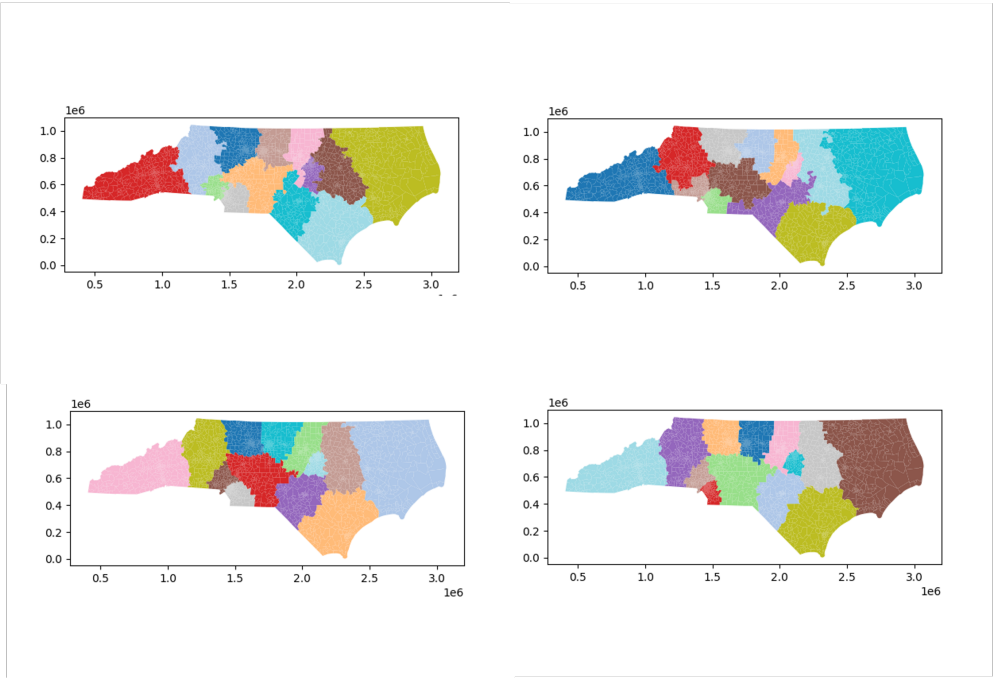


Figure 4.7: NC medoids, each column is for a specific seed. Top row: A_{closest}, Bottom row: $\hat{A}^*$.

allow transitions to new states (maps) that are closer to the centroid and we do this for a total of 200,000 steps to obtain the final estimated medoid $\hat{A}^*$[8]. We follow this procedure three times one for each centroid [9]. Figure 4.7 (top row) shows the A_{closest} medoids from two different runs (each comparing to a different centroid). It is not difficult to see that they are different. The bottom row shows the final medoids after we run the chain from A_{closest} to obtain $\hat{A}^*$. We see that the final medoids are indeed very similar and in fact when we measure the distances between them we find them to be very close.

[8] Similair to the centroid estimation, we also use a smaller number of sample for for PA and MD but we find that we get similiar results empirically.

[9] As mentioned before we get three centroids each from sampling a chain that starts with a different seed.

Chapter 5: Remarks and Future Work

In this chapter we give some remarks and possible opportunities for future work. We will focus on fair clustering in section 5.1 and on redistricting and gerrymandering in section 5.2.

5.1 Fair Clustering

Fair clustering has received significant attention and as stated in section 2.5 we can count at least seven different fairness notions. However, it is worthwhile to wonder if the introduction of so many constraints is an advantage and a sign of progress in the field? Each fairness notion (while being well-motivated) is introduced in a manner that ignores the previously introduced notions. It seems reasonable that a decision maker wanting to ensure fairness in a clustering-centered application is faced with a non-trivial problem where he has to pick a constraint and justify picking it (possibly over other constraints[1]).

Further, our work in section 2.5 shows how to satisfy two fairness notions in clustering simultaneously as well as incompatibility results between some fairness notions in clustering. One can envision other projects on this kind where multiple fairness notions are considered simultaneously. This would lead to a deeper understanding of the interaction between different fairness notions. However, it is also worthwhile to wonder if such an approach is scalable? An

[1] In section 2.5 we showed that some constraints have in general an empty feasible set, i.e. they are incompatible.

approach that is simple and direct is arguably much preferred both from a theoretical and practical point of view. For example, the work of [151–153] suggests a welfare-based approach to algorithmic fairness. Such an approach seems more suited and possibly preferred over the current approach of introducing constraints but not describing the welfare interaction between the individuals/demographic groups and the algorithm.

Another remark concerns the probabilistic fairness notion which was introduced in section 2.2. While the issue of imperfect knowledge of group memberships is well-motivated and common. The introduced notion of fairness in expectation is arguably lacking. Specifically, it is more preferable for the clustering output to be fair in realization or with high probability instead of being fair in expectation. Further, the model assumes that the probability of a point belonging to a group is known (for all points and groups) which might be a strong assumption in many applications. A simpler model with weaker assumptions and an algorithm which achieves fairness deterministically are certainly preferred.

5.2 Redistricting and Gerrymandering

Computational methods in redistricting and gerrymandering have received significant attention in recent years and as discussed in chapter 4 have led to impactful results. The main engine behind this is the MCMC methods such as those of [32, 33]. It is possible that these methods have an even greater potential which has not been utilized yet. Specifically, if we think of the ensemble of redistricting maps resulting from an MCMC method as a training dataset in machine learning. For example, we can apply high dimensional visualization methods such as those of [154] to gain a greater insight into the structure of the dataset (ensemble of maps). Fur-

ther, such a visualization method can be used to give a simpler and more direct argument for why a specific map should be used or why another map is likely to be gerrymandered. Another possibility, is using anomaly detection methods [155] to possibly detect gerrymandered maps. In fact, our work in chapter 4 closely resembles some methods in anomaly detection.

Bibliography

[1] Ioana O Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R Schmidt, and Melanie Schmidt. On the cost of essentially fair clusterings. 2019.

[2] Timothy P Cadigan and Christopher T Lowenkamp. Implementing risk assessment in the federal pretrial services system. *Fed. Probation*, 75:30, 2011.

[3] 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.

[4] Martin Leo, Suneel Sharma, and Koilakuntla Maddulety. Machine learning in banking risk management: A literature review. *Risks*, 7(1):29, 2019.

[5] Miranda Bogen and Aaron Rieke. Help wanted: An examination of hiring algorithms, equity, and bias. 2018.

[6] Michael Kearns and Aaron Roth. *The ethical algorithm: The science of socially aware algorithm design*. Oxford University Press, 2019.

[7] Cathy O'neil. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Broadway Books, 2016.

[8] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems 30*, 2017.

[9] 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*, pages 259–268, 2015.

[10] Serena Wang, Wenshuo Guo, Harikrishna Narasimhan, Andrew Cotter, Maya Gupta, and Michael Jordan. Robust optimization for fairness with noisy protected groups. *Advances in Neural Information Processing Systems*, 33:5190–5203, 2020.

[11] Pranjal Awasthi, Matthäus Kleindessner, and Jamie Morgenstern. Equalized odds post-processing under imperfect group information. In *International Conference on Artificial Intelligence and Statistics*, pages 1770–1780. PMLR, 2020.

[12] Flavien Prost, Pranjal Awasthi, Nick Blumm, Aditee Kumthekar, Trevor Potter, Li Wei, Xuezhi Wang, Ed H Chi, Jilin Chen, and Alex Beutel. Measuring model fairness under noisy covariates: A theoretical perspective. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 873–883, 2021.

[13] Pranjal Awasthi, Alex Beutel, Matthäus Kleindessner, Jamie Morgenstern, and Xuezhi Wang. Evaluating fairness of machine learning models under uncertain and incomplete information. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 206–214, 2021.

[14] Seyed Esmaeili, Brian Brubach, Leonidas Tsepenekas, and John Dickerson. Probabilistic fair clustering. *Advances in Neural Information Processing Systems*, 33:12743–12755, 2020.

[15] Dimitris Bertsimas, Vivek F Farias, and Nikolaos Trichakis. The price of fairness. *Operations research*, 59(1):17–31, 2011.

[16] John P Dickerson, Ariel D Procaccia, and Tuomas Sandholm. Price of fairness in kidney exchange. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 2014.

[17] Seyed Esmaeili, Brian Brubach, Aravind Srinivasan, and John Dickerson. Fair clustering under a bounded cost. *Advances in Neural Information Processing Systems*, 34, 2021.

[18] Ulrike Von Luxburg, Robert C Williamson, and Isabelle Guyon. Clustering: Science or art? In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 65–79. JMLR Workshop and Conference Proceedings, 2012.

[19] 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.

[20] Awasthi Pranjal, 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 Conference on Artificial Intelligence (AAAI)*, 2022.

[21] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Neural Information Processing Systems*, 2017.

[22] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. 2019.

[23] Huy Lê Nguyen, Thy Nguyen, and Matthew Jones. Fair range k-center. *arXiv preprint arXiv:2207.11337*, 2022.

[24] John Dickerson, Seyed Esmaeili, Jamie Morgenstern, and Claire Jie Zhang. Doubly constrained fair clustering. *arXiv preprint arXiv:2305.19475*, 2023.

[25] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for online bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358, 1990.

[26] Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.

[27] 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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2210–2217, 2020.

[28] Yifan Xu and Pan Xu. Trading the system efficiency for the income equality of drivers in rideshare. *arXiv preprint arXiv:2012.06850*, 2020.

[29] Will Ma, Pan Xu, and Yifan Xu. Group-level fairness maximization in online bipartite matching. *arXiv preprint arXiv:2011.13908*, 2020.

[30] Seyed A Esmaeili, Sharmila Duppala, Davidson Cheng, Vedant Nanda, Aravind Srinivasan, and John P Dickerson. Rawlsian fairness in online bipartite matching: Two-sided, group, and individual. *AAAI*, 2023.

[31] William Vickrey. On the prevention of gerrymandering. *Political Science Quarterly*, 76(1):105–110, 1961.

[32] Daryl DeFord, Moon Duchin, and Justin Solomon. Recombination: A family of markov chains for redistricting. *arXiv preprint arXiv:1911.05725*, 2019.

[33] Jonathan C Mattingly and Christy Vaughn. Redistricting and the will of the people. *arXiv preprint arXiv:1410.8796*, 2014.

[34] League of Women Voters of Pennsylvania v. Commonwealth of Pennsylvania, No. 159 MM (2018).

[35] Seyed A Esmaeili, Darshan Chakrabarti, Hayley Grape, and Brian Brubach. Implications of distance over redistricting maps: Central and outlier maps. *arXiv preprint arXiv:2203.00872*, 2023.

[36] Gagan Aggarwal, Rina Panigrahy, Tomás Feder, Dilys Thomas, Krishnaram Kenthapadi, Samir Khuller, and An Zhu. Achieving anonymity via clustering. *ACM Transactions on Algorithms (TALG)*, 6(3):49, 2010.

[37] Samir Khuller and Yoram J Sussmann. The capacitated k-center problem. *SIAM Journal on Discrete Mathematics*, 13(3):403–418, 2000.

[38] Marek Cygan, MohammadTaghi Hajiaghayi, and Samir Khuller. Lp rounding for k-centers with non-uniform hard capacities. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 273–282. IEEE, 2012.

[39] Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. Clustering without over-representation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 267–275, 2019.

[40] Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. *Advances in Neural Information Processing Systems*, 32, 2019.

[41] Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical computer science*, 38:293–306, 1985.

[42] Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 737–756. SIAM, 2014.

[43] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal on computing*, 33(3):544–562, 2004.

[44] D Arthur and S Vassilvitskii. k-means++: The advantages of careful seeding. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, 2007.

[45] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

[46] Pranjal Awasthi, Matthäus Kleindessner, and Jamie Morgenstern. Effectiveness of equalized odds for fair classification under imperfect group information. *arXiv preprint arXiv:1906.03284*, 2019.

[47] Christopher Meek, Bo Thiesson, and David Heckerman. The learning-curve sampling method applied to model-based clustering. *Journal of Machine Learning Research*, 2(Feb):397–418, 2002.

[48] Dimitris Bertsimas, Vivek F Farias, and Nikolaos Trichakis. The price of fairness. 59(1):17–31, 2011.

[49] Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, and Maria Kyropoulou. The efficiency of fair division. International Workshop on Internet and Network Economics (WINE), 2009.

[50] John P. Dickerson, Ariel D. Procaccia, and Tuomas Sandholm. Price of fairness in kidney exchange. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1013–1020, 2014.

[51] United States Senate. S. 1745 – 102nd Congress: Civil Rights Act of 199, 1991. https://www.govtrack.us/congress/bills/102/s1745.

[52] Supreme Court of the United States. 13-1371 – Texas Department of Housing and Community Affairs v. The Inclusive Communities Project, Inc., January 2015.

[53] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of computational social choice*. Cambridge University Press, 2016.

[54] G Gunduz and Ernest Fokoue. Uci machine learning repository. *Irvine, CA: University of California, School of Information and Computer Science*, 20, 2013.

[55] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *ACM Conference on Fairness, Accountability, and Transparency (ACM FAccT)*, pages 77–91, 2018.

[56] 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. 2012.

[57] 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.

[58] Pang-Ning Tan, Michael Steinbach, DA Karpatne, and DV Kumar. Introduction to data mining , 2nd editio, 2018.

[59] 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.

[60] Ava Kofman and Ariana Tobin. Facebook ads can still discriminate against women and older workers, despite a civil rights settlement. 2019.

[61] 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 *ACM Conference on Fairness, Accountability, and Transparency*, 2018.

[62] Amit Datta, Anupam Datta, Jael Makagon, Deirdre K Mulligan, and Michael Carl Tschantz. Discrimination in online advertising: A multidisciplinary inquiry. In *ACM Conference on Fairness, Accountability, and Transparency*, 2018.

[63] Deepak P. Whither fair clustering? In *AI for Social Good Workshop*, 2020.

[64] 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.

[65] Dachuan Xu and Shuzhong Zhang. Approximation algorithm for facility location with service installation costs. *Operations Research Letters*, 36(1):46–50, 2008.

[66] Ian Davidson and SS Ravi. Making existing clusterings fairer: Algorithms, complexity results and insights. In *AAAI Conference on Artificial Intelligence*, 2020.

[67] Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

[68] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 5. Springer, 2015.

[69] 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.

[70] 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. 2011.

[71] Dheeru Dua and Casey Graff. Uci machine learning repository. 2017.

[72] Lingxiao Huang, Shaofeng Jiang, and Nisheeth Vishnoi. Coresets for clustering with fairness constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7589–7600, 2019.

[73] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. Scalable fair clustering. 2019.

[74] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

[75] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

[76] Matthew Jones, Huy Lê Nguyên, and Thy Nguyen. Fair k-centers via maximum matching. In *International Conference on Machine Learning (ICML)*, 2020.

[77] Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. Fair clustering via equitable group representations, 2020.

[78] 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.

[79] Christopher Jung, Sampath Kannan, and Neil Lutz. A center in your neighborhood: Fairness in facility location. 2019.

[80] Xingyu Chen, Brandon Fain, Charles Lyu, and Kamesh Munagala. Proportionally fair clustering. 2019.

[81] 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.

[82] Aric Hagberg, Dan Schult, Pieter Swart, D Conway, L Séguin-Charbonneau, C Ellison, B Edwards, and J Torrents. Networkx. high productivity software for complex networks. *https://networkx.github.io/*, 2013.

[83] A Frank. Uci machine learning repository. irvine, ca: University of california, school of information and computer science. *http://archive. ics. uci. edu/ml*, 2010.

[84] Chien-Ju Ho and Jennifer Vaughan. Online task assignment in crowdsourcing markets. In *AAAI*, 2012.

[85] Yongxin Tong, Jieying She, Bolin Ding, Libin Wang, and Lei Chen. Online mobile micro-task allocation in spatial crowdsourcing. In *ICDE*, 2016.

[86] John P Dickerson, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. Balancing relevance and diversity in online bipartite matching via submodularity. In *AAAI*, 2019.

[87] Meghna Lowalekar, Pradeep Varakantham, and Patrick Jaillet. Online spatio-temporal matching in stochastic and dynamic domains. *Artificial Intelligence (AIJ)*, 261:71–112, 2018.

[88] 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.

[89] Will Ma, Pan Xu, and Yifan Xu. Fairness maximization among offline agents in online-matching markets. In *WINE*, 2021.

[90] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 982–991, 2008.

[91] Gina Cook. Woman says uber driver denied her ride because of her wheelchair. 2018.

[92] Gillan B. White. Uber and lyft are failing black riders. 2016.

[93] Eli Wirtschafter. Driver discrimination still a problem as uber and lyft prepare to go public. 2019.

[94] Alex Rosenblat, Karen EC Levy, Solon Barocas, and Tim Hwang. Discriminating tastes: Customer ratings as vehicles for bias. *Available at SSRN 2858946*, 2016.

[95] Hernan Galperin and Catrihel Greppi. Geographical discrimination in the gig economy. *Available at SSRN 2922874*, 2017.

[96] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

[97] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science Conference*, 2012.

[98] Nixie S Lesmana, Xuan Zhang, and Xiaohui Bei. Balancing efficiency and fairness in on-demand ridesourcing. In *NeurIPS*, 2019.

[99] Will Ma and Pan Xu. Group-level fairness maximization in online bipartite matching. In *AAMAS*, 2022.

[100] Yifan Xu and Pan Xu. Trade the system efficiency for the income equality of drivers in rideshare. In *IJCAI*, 2020.

[101] 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.

[102] 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.

[103] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *Proceedings of The Web Conference 2020*, pages 1194–1204, 2020.

[104] Kinjal Basu, Cyrus DiCiccio, Heloise Logan, and Noureddine El Karoui. A framework for fairness in two-sided marketplaces. *arXiv preprint arXiv:2006.12756*, 2020.

[105] John Rawls. Justice as fairness. *The philosophical review*, 67(2):164–194, 1958.

[106] David García-Soriano and Francesco Bonchi. Fair-by-design matching. *Data Mining and Knowledge Discovery*, pages 1–45, 2020.

[107] 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.

[108] Nicholas Mattei, Abdallah Saffidine, and Toby Walsh. Mechanisms for online organ matching. In *IJCAI*, 2017.

[109] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, , and Ariel D Procaccia. WeBuildAI: Participatory framework for algorithmic governance. In *CSCW*, 2019.

[110] Quan Zhou, Jakub Marecek, and Robert N Shorten. Subgroup fairness in two-sided markets. *arXiv preprint arXiv:2106.02702*, 2021.

[111] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e, 2009.

[112] 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.

[113] 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.

[114] Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. Online stochastic matching: New algorithms and bounds, 2016.

[115] Marek Adamczyk, Fabrizio Grandoni, and Joydeep Mukherjee. Improved approximation algorithms for stochastic matching. In *Algorithms-ESA 2015*, pages 1–12. Springer, 2015.

[116] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[117] 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.

[118] 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.

[119] 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.

[120] Maria Chikina, Alan Frieze, and Wesley Pegden. Assessing significance in a markov chain without mixing. *Proceedings of the National Academy of Sciences*, 114(11):2860–2864, 2017.

[121] Gregory Herschlag, Han Sung Kang, Justin Luo, Christy Vaughn Graves, Sachet Bangia, Robert Ravier, and Jonathan C Mattingly. Quantifying gerrymandering in north carolina. *Statistics and Public Policy*, 7(1):30–38, 2020.

[122] Rucho v. Common Cause, No. 18-422, 588 U.S. ___ (2019).

[123] Matt Chen. Tufts research lab aids states with redistricting process. *The Tufts Daily*, April 6, 2021.

[124] Vincent Cohen-Addad, Philip N Klein, and Neal E Young. Balanced centroidal power diagrams for redistricting. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 389–396, 2018.

[125] Cyrus Hettle, Shixiang Zhu, Swati Gupta, and Yao Xie. Balanced districting on grid graphs with provable compactness and contiguity. *arXiv preprint arXiv:2102.05028*, 2021.

[126] Yan Y. Liu, Wendy K. Tam Cho, and Shaowen Wang. Pear: a massively parallel evolutionary computation approach for political redistricting optimization and analysis. *Swarm and Evolutionary Computation*, 30:78 – 92, 2016.

[127] John G Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[128] Lucy Chenyun Wu, Jason Xiaotian Dou, Danny Sleator, Alan Frieze, and David Miller. Impartial redistricting: A markov chain approach. *arXiv preprint arXiv:1510.03247*, 2015.

[129] Benjamin Fifield, Michael Higgins, Kosuke Imai, and Alexander Tarr. A new automated redistricting simulator using markov chain monte carlo. *Work. Pap., Princeton Univ., Princeton, NJ*, 2015.

[130] Brian Brubach, Aravind Srinivasan, and Shawn Zhao. Meddling metrics: the effects of measuring and constraining partisan gerrymandering on voter incentives. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 815–833, 2020.

[131] Aaron Bycoffe, Ella Koeze, David Wasserman, and Julia Wolfe. The Atlas Of Redistricting. `https://projects.fivethirtyeight.com/redistricting-maps/`, 2018. [Online; published 25-January-2018; accessed 15-August-2019].

[132] Daniel D Polsby and Robert D Popper. The third criterion: Compactness as a procedural safeguard against partisan gerrymandering. *Yale Law & Policy Review*, 9(2):301–353, 1991.

[133] Wesley Pegden, Ariel D Procaccia, and Dingli Yu. A partisan districting protocol with provably nonpartisan outcomes. *arXiv preprint arXiv:1710.08781*, 2017.

[134] Ivan Ryan and Warren D. Smith. Splitline districtings of all 50 states + DC + PR. `https://rangevoting.org/SplitLR.html`. [Online; accessed 15-August-2019].

[135] Wendy K Tam Cho. Technology-enabled coin flips for judging partisan gerrymandering. *Southern California law review*, 93, 2019.

[136] Vieth v. Jubelirer, No. 02-1580, 541 U.S. 267 (2004).

[137] Tara Abrishami, Nestor Guillen, Parker Rule, Zachary Schutzman, Justin Solomon, Thomas Weighill, and Si Wu. Geometry of graph partitions via optimal transport. *SIAM Journal on Scientific Computing*, 42(5):A3340–A3366, 2020.

[138] Boris Alexeev and Dustin G Mixon. An impossibility theorem for gerrymandering. *The American Mathematical Monthly*, 125(10):878–884, 2018.

[139] Anshuman Chhabra, Abhishek Roy, and Prasant Mohapatra. Suspicion-free adversarial attacks on clustering algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3625–3632, 2020.

[140] Antonio Emanuele Cinà, Alessandro Torcinovich, and Marcello Pelillo. A black-box adversarial attack for poisoning clustering. *Pattern Recognition*, 122:108306, 2022.

[141] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[142] Maryland Citizens Redistricting Commission. Redistricting Map Submission Process. `https://redistricting.maryland.gov/Pages/plan-proposals.aspx`, 2021. [Online; accessed 20-October-2021].

[143] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern recognition letters*, 24(9-10):1641–1650, 2003.

[144] Edwin M Knox and Raymond T Ng. Algorithms for mining distancebased outliers in large datasets. In *Proceedings of the international conference on very large data bases*, pages 392–403. Citeseer, 1998.

[145] James Newling and François Fleuret. A sub-quadratic exact medoid algorithm. In *Artificial Intelligence and Statistics*, pages 185–193. PMLR, 2017.

[146] Stefan Zubrzycki. *Lectures in probability theory and mathematical statistics*, volume 38. Elsevier Publishing Company, 1972.

[147] Olivier Goldschmidt and Dorit S Hochbaum. A polynomial algorithm for the k-cut problem for fixed k. *Mathematics of operations research*, 19(1):24–37, 1994.

[148] Huzur Saran and Vijay V Vazirani. Finding k cuts within twice the optimal. *SIAM Journal on Computing*, 24(1):101–108, 1995.

[149] Hamidreza Validi and Austin Buchanan. Political districting to minimize cut edges. *Mathematical Programming Computation*, pages 1–50, 2022.

[150] Daryl DeFord and Moon Duchin. Redistricting reform in virginia: Districting criteria in context. *Virginia Policy Review*, 12(2):120–146, 2019.

[151] Violet Xinying Chen and JN Hooker. Fairness through social welfare optimization. *arXiv preprint arXiv:2102.00311*, 2021.

[152] Lily Hu and Yiling Chen. Fair classification and social welfare. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 535–545, 2020.

[153] Alex Chohlas-Wood, Madison Coots, Henry Zhu, Emma Brunskill, and Sharad Goel. Learning to be fair: A consequentialist approach to equitable decision-making. *arXiv preprint arXiv:2109.08792*, 2021.

[154] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[155] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.