ABSTRACT<br>\title{ of Dissertation: ON ALGORITHMIC FAIRNESS AND STOCHASTIC MODELS FOR COMBINATORIAL OPTIMIZATION AND UNSUPERVISED MACHINE LEARNING }<br>Leonidas Tsepenekas<br>Doctor of Philosophy, 2022<br>Dissertation Directed by: Professor Aravind Srinivasan, Department of Computer Science

Combinatorial optimization and unsupervised machine learning problems have been extensively studied and are relatively well-understood. Examples of such problems that play a central role in this work are clustering problems and problems of finding cuts in graphs. The goal of the research presented in this dissertation is to introduce novel variants of the aforementioned problems, by generalizing their classic variants into two, not necessarily disjoint, directions. The first direction involves incorporating fairness aspects to a problem's specifications, and the second involves the introduction of some form of randomness in the problem definition, e.g., stochastic uncertainty about the problem's parameters.

Fairness in the design of algorithms and in machine learning has received a significant amount of attention during the last few years, mainly due to the realization that standard optimization approaches can frequently lead to severely unfair outcomes, that can potentially hurt the individuals or the groups involved in the corresponding application. As far as considerations
of fairness are concerned, in this work we begin by presenting two novel individually-fair clustering models, together with algorithms with provable guarantees for them. The first such model exploits randomness in order to provide fair solutions, while the second is purely deterministic. The high-level motivation behind both of them is trying to treat similar individuals similarly. Moving forward, we focus on a graph cut problem that captures situations of disaster containment in a network. For this problem we introduce two novel fair variants. The first variant focuses on demographic fairness, while the second considers a probabilistic notion of individual fairness. Again, we give algorithms with provable guarantees for the newly introduced variants.

In the next part of this thesis we turn our attention to generalizing problems through the introduction of stochasticity. At first, we present algorithmic results for a computational epidemiology problem, whose goal is to control the stochastic diffusion of a disease in a contact network. This problem can be interpreted as a stochastic generalization of a static graph cut problem. Finally, this dissertation also includes work on a well-known paradigm in stochastic optimization, namely the two-stage stochastic setting with recourse. Two-stage problems capture a wide variety of applications revolving around the trade-off between provisioning and rapid response. In this setting, we present a family of clustering problems that had not yet been studied in the literature, and for this family we show novel algorithmic techniques that provide constant factor approximation algorithms.

We conclude the dissertation with a discussion on open problems and future research directions in the general area of algorithmic fairness.

# ON ALGORITHMIC FAIRNESS AND STOCHASTIC MODELS FOR COMBINATORIAL OPTIMIZATION AND UNSUPERVISED MACHINE LEARNING 

by

Leonidas Tsepenekas

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
2022

Advisory Committee:
Professor Aravind Srinivasan, Chair/Advisor
Professor John P. Dickerson, Co-Advisor
Professor Shuvra S. Bhattacharyya
Professor Hal Daume
Professor Leonidas Lampropoulos
Dr. Ravi Kumar
© Copyright by
Leonidas Tsepenekas

In memory of Niki Tsepeneka.

## Acknowledgments

First of all, I would like to sincerely thank my academic advisors, professors John P. Dickerson and Aravind Srinivasan. Without their guidance, help and patience this work would not have been possible. Aravind has been there for me since the day I joined UMD, and it was a privilege working with such a brilliant and kind person. I also feel extremely grateful for John agreeing to be my co-advisor at the end of my third year. His excitement and his insight on real world problems fueled my passion for research and opened new horizons to my work.

I would also like to thank my mentors during my two summer internships. Anil Vullikanti welcomed me to the Biocomplexity Institute at the University of Virginia, and he introduced me to the exciting area of computational epidemiology. Further, I am grateful to Ivan Brugere for an amazingly fruitful summer working together at JPMorgan Chase \& Co. I am looking forward to continuing this collaboration as colleagues.

I gratefully acknowledge NSF awards CCF-1749864 and CCF-1918749 as well as research awards from Amazon and Google for supporting my research. In addition, I would also like to thank all my other precious co-authors for their valuable contributions: Brian Brubach, Darshan Chakrabarti, Michael Dinitz, Seyed A. Esmaeili, Nathaniel Grammel, David G. Harris, George Li, Ann Li, Madhav Marathe and Samir Khuller.

I'm also grateful to Professors Shuvra S. Bhattacharyya, Hal Daume, Leonidas Lampropoulos and Dr. Ravi Kumar for agreeing to participate in my committee.

Of course, I cannot forget all the friends I made the last 5 years; without the "Greek" squad I would not have been able to push through all the stress and the difficulties of the PhD life. Special thanks to Eleftheria, Leo \#2 (I'm sorry, but I was the first to arrive in College Park), Joho, and the best group of roommates: Papa, Panos, Chris, George ("Den lhgei o traxanas!")

Last but not least, I cannot describe how grateful I am to my family. At first, my parents and my sister with their tremendous support and their unconditional love. Then, my dearest Melissa who has been one of my main driving factors; thank you for your support, your advice and your patience. Lastly, I have to acknowledge little Athena! Your smiles and your laughter are priceless. Hopefully, once this thesis is defended, dad will be more available for playtime!

## Table of Contents

Dedication ..... ii
Acknowledgements ..... iii
Table of Contents ..... V
List of Tables ..... vii
List of Figures ..... viii
Chapter 1: Introduction ..... 1
1.1 Brief Introduction to Clustering Problems ..... 2
1.2 Brief Introduction to Graph-Cut Problems ..... 4
1.3 Formal Problem Definitions and Motivation ..... 4
1.3.1 Individually-Fair Clustering ..... 5
1.3.2 Fair Graph-Cut Problems ..... 11
1.3.3 A Computational Epidemiology Problem ..... 15
1.3.4 Two-Stage Stochastic Clustering ..... 18
1.4 Contributions and Outline ..... 22
1.4.1 Contributions to PairFairclu ..... 22
1.4.2 Contributions to EQCEnter ..... 25
1.4.3 Contributions to Fair Graph-Cut Problems ..... 29
1.4.4 Contributions to Computational Epidemiology ..... 31
1.4.5 Contributions to Two-Stage Stochastic Clustering ..... 33
1.5 Literature Review ..... 38
1.5.1 Fair Clustering Literature ..... 38
1.5.2 Stochastic Clustering Literature ..... 43
1.5.3 Vanilla Clustering Literature ..... 43
1.5.4 Graph-Cuts and Computational Epidemiology Literature ..... 43
1.6 Pointers to the Most Important Results of the Thesis ..... 44
Chapter 2: Addressing Individually Fair Clustering ..... 45
2.1 Solving PairFairClu ..... 45
2.1.1 Some First Simple Results ..... 45
2.1.2 More Refined Results for PairFairclu ..... 48
2.1.3 Experimental Results for PairFairclu ..... 58
2.2 Solving EQCENTER ..... 63
2.2.1 Structural Properties of the Problem ..... 63
2.2.2 Approximation Algorithms for EQCenter-PP and EQCenter-AG ..... 73
2.2.3 Solving the Assignment Problem ..... 86
2.2.4 Experimental Evaluation for EQCENTER ..... 89
Chapter 3: Addressing the Fair Cuts Problems ..... 94
3.1 Reduction to Tree Instances ..... 94
3.2 Solving DemFairCut ..... 98
3.2.1 Solving DemFairCut for $\gamma=O(1)$ ..... 98
3.2.2 Solving DemFairCut for an Arbitrary $\gamma$ ..... 99
3.3 Solving IndFairCut ..... 108
3.3.1 A $(1,1, O(\log n)+\epsilon)$-Bicriteria Algorithm for AuxCut ..... 108
3.3.2 A Round-or-Cut Solution for IndFairCut ..... 110
Chapter 4: Mitigating the Stochastic Spread of A Disease ..... 114
4.1 MinInfEdge with Unit Edge-Costs and Uniform Probabilities ..... 114
4.2 The SAA Path-Dependent Framework for Arbitrary Networks ..... 117
4.3 Counting Paths in the Chung-Lu Random Graph Model ..... 124
4.3.1 A Positive Result When $\beta>3$ ..... 126
4.3.2 A Negative Result When $\beta<3$ ..... 129
4.4 A Deterministic Rounding SAA Approach ..... 130
Chapter 5: Addressing Two-Stage Stochastic Clustering ..... 134
5.1 Notation and Important Subroutines ..... 134
5.2 Generalizing to the Black-Box Setting ..... 135
5.3 Solving 2S-Sup-Poly ..... 146
5.3.1 A 3-Approximation Algorithm for 2S-Sup-Poly ..... 146
5.3.2 Generalizing to the Black-Box Setting ..... 148
5.4 Solving 2S-MATSuP-Poly ..... 150
5.4.1 A 5-Approximation Algorithm for 2S-MATSuP-Poly ..... 150
5.4.2 Generalizing to the Black-Box Setting ..... 153
5.5 Solving 2S-MuSup-Poly ..... 154
5.5.1 Reducing 2S-MuSup-Poly to Robust Weighted Multi-Knapsack-Sup ..... 155
5.5.2 Generalizing to the Black-Box Setting ..... 158
5.5.3 Connections to 2S-MATSup-Poly ..... 160
5.6 Applying the Standard SAA Method in Supplier Problems ..... 161
5.6.1 The Standard SAA Method: ..... 161
5.6.2 Roadblocks for the Standard SAA Analysis in Supplier Problems ..... 162
Chapter 6: Conclusions and Future Work ..... 164
6.1 Future Work ..... 165
Appendix A: Additional Experimental Results for EQCENTER ..... 168
Bibliography ..... 172

## List of Tables

2.1 Percentage of constraints that are violated on average for metric $F_{2}$ ..... 60
2.2 Percentage of constraints that are violated on average for metric $F_{3}$ ..... 61
2.3 Cost of fairness for metric $F_{2}$ ..... 62
2.4 Cost of fairness for metric $F_{3}$ ..... 62
2.5 Percentage of constraints that are violated on average for metric $F_{1}$ ..... 63
2.6 Objective achieved for metric $F_{1}$ ..... 63

## List of Figures

2.1 Construction for $m=8$. The solid lines represent a distance of 1 between the cor- responding points. The dashed lines correspond to similarity sets. For instance, the dashed line between $j_{1}$ and $j_{9}$ shows that $\pi\left(j_{1}\right)=j_{9}$ and $\pi\left(j_{9}\right)=j_{1}$. . . . . 64
2.2 PoF example: Lines represent distances, such that $d\left(j_{1}, j_{2}\right)=R^{\prime}, d\left(j_{2}, j_{3}\right)=R$. ..... 71
2.3 Adult: Maximum assignment distance for all algorithms ..... 91
2.4 Adult: Satisfaction of fairness constraints ..... 92
2.5 Adult: Amount of constraint violation ..... 93
A. 1 Bank: Maximum assignment distance for all algorithms ..... 168
A. 2 Bank: Satisfaction of fairness constraints ..... 168
A. 3 Bank: Amount of constraint violation ..... 168
A. 4 Credit: Maximum assignment distance for all algorithms ..... 169
A. 5 Credit: Satisfaction of fairness constraints ..... 169
A. 6 Credit: Amount of constraint violation ..... 169
A. 7 Census: Maximum assignment distance for all algorithms ..... 170
A. 8 Census: Satisfaction of fairness constraints ..... 170
A. 9 Census: Amount of constraint violation ..... 170
A. 10 Diabetes: Maximum assignment distance for all algorithms ..... 171
A. 11 Diabetes: Satisfaction of fairness constraints ..... 171
A. 12 Diabetes: Amount of constraint violation ..... 171

## Chapter 1: Introduction

Combinatorial optimization and unsupervised machine learning problems are fundamental to theoretical Computer Science, they have been studied for many decades now, and therefore they are relatively well-understood. The purpose of this thesis is to generalize a family of such classical problems into two, not necessarily disjoint, vital directions. The first is considering fair variants of those problems; the second involves studying stochastic generalizations of them.

The study of fairness in computer science, game theory and economics dates back many decades. However, it is only in recent years that fairness in algorithmic design and machine learning has received a significant amount of attention [1, 2]. This occurred mainly due to the realization that the output of standard optimization algorithms which are used on a daily basis can very well lead to outcomes that are highly unfair and hurtful for the individuals or the groups involved. Examples of this include among others racial bias in Airbnb rentals [3], gender bias in Google's Ad Settings [4] and discrimination in housing ads on Facebook [5]. There are two reasons why such unfortunate events occur. First, the training datasets used include implicit biases, and hence when algorithms are trained on them, they learn to perpetuate the underlying biases. Second, in many situations, even if the data is completely unbiased, merely optimizing an objective function does not suffice if fairness considerations are at play. In such cases, we must explicitly incorporate fairness constraints in our algorithm design process. Our work here tries
to accomplish the latter. Finally, with automated decisions becoming ubiquitous in critical areas affecting the well-being of human lives, e.g., issuing home loans via estimating credit scores, predicting recidivism and computing risk factors for health insurance applicants, it is evident that the design of fair algorithms that minimize bias and discrimination is of vital importance.

Regarding the second research direction we pursue, i.e., considering stochastic variants of classical problems, we study randomness that comes in two flavors. At first, we consider situations where the input of the problem is not deterministically known, and we only have stochastic knowledge of it. Such cases naturally model uncertainty in the realization of the input data. In addition, we consider variants of classical problems where the solution itself needs to be randomized. In other words, we study problems where the solution needs to be a distribution over deterministic solutions. The benefit of such randomized solutions is that they allow us to impose stochastic constraints on the problem, and we will demonstrate situations where this is very helpful, in the sense of capturing real-life scenarios. Specifically, we present examples where such a model has the potential of introducing fairness features to the problem. This interplay of stochasticity and fairness, with emphasis on how the latter can enforce fair outcomes, is one of the most intriguing aspects of this work.

As for the exact families of classical problems that we are interested in generalizing, we focus on clustering and graph-cut problems.

### 1.1 Brief Introduction to Clustering Problems

Clustering is one of the most popular paradigms at the intersection of combinatorial optimization and unsupervised machine learning. In a standard clustering problem, there is a
set of points $\mathcal{C}$ and a set of locations $\mathcal{F}$ in a metric space characterized by a distance function $d:(\mathcal{C} \cup \mathcal{F})^{2} \mapsto \mathbb{R}_{\geq 0} .^{1}$ In addition, the input also includes a positive integer $k$. The goal is to find a set of centers $S \subseteq \mathcal{F}$ with $|S| \leq k$, and subsequently construct an assignment $\phi: \mathcal{C} \mapsto S$ that maps each point to one of the chosen centers, thus creating a collection of $k$ clusters. Moreover, the quantity that really matters for each $j \in \mathcal{C}$, is its distance $d(j, \phi(j))$ to its corresponding cluster center $\phi(j)$. In classical clustering applications $d(j, \phi(j))$ would correspond to how similar $\phi(j)$ is to $j$, and in facility-location applications to the distance $j$ needs to travel in order to reach its service-provider. In general, the smaller $d(j, \phi(j))$ is the happier $j$.

The most popular objectives in the literature ( $k$-center, $k$-median, $k$-means) "boil down" the large collection of values $d(j, \phi(j))$, into an increasing function they try to minimize.

1. $k$-center objective: Minimize $\max _{j \in \mathcal{C}} d(j, \phi(j))$.
2. $k$-median objective: Minimize $\sum_{j \in \mathcal{C}} d(j, \phi(j))$.
3. $k$-means objective: Minimize $\sum_{j \in \mathcal{C}} d(j, \phi(j))^{2}$.

The $k$-center objective can be viewed as trying to minimize the worst case scenario over all points, while the other two can be interpreted as minimizing the average "service" points receive.

There are two distinct motivational settings for clustering problems. In the first, these problems are used to model situations where we want to group together points that are highly similar (assuming $\mathcal{C}=\mathcal{F}$ ); here the cluster centers are thought of as representative points that can define a relatively homogeneous group and similarity is measured through the distance function $d$ (points that are closer together are deemed more similar). The second motivational setting

[^0]comes from operations research, and specifically from facility location applications. For this case, centers correspond to facilities that are placed in a metric, such that the clients of the set $\mathcal{C}$ are served in a distance efficient way.

### 1.2 Brief Introduction to Graph-Cut Problems

Problems on graphs are among the most well-studied settings in combinatorial optimization. An extremely important family of such problems are problems of finding cuts in an graph. In such a setting, we are given a graph $G=(V, E)$ with vertex set $V$ and edge set $E$, where each edge $e \in E$ has some non-negative weight $w_{e} \geq 0$. On a high level, the goal is to remove a set of edges $F \subseteq E$ from the graph (this set is called the cut-set or simply cut), such that some connectivity constraints are satisfied in the residual graph $G_{F}=(V, E \backslash F)$, and the cost $w(F)^{2}$ of the cut satisfies certain properties, e.g., it is minimized. As it will later become evident through specific examples, graph-cuts have numerous applications in network problems, ranging from epidemiology to computer and social networks analysis.

### 1.3 Formal Problem Definitions and Motivation

In this section we are going to define and motivate all problems of interest that this work addresses. We begin by exploring two individually-fair clustering models. Then we move on to two fair graph-cut problems. We next introduce a computational epidemiology problem that constitutes a stochastic variant of a graph-cut problem. Finally we define a clustering problem in the two-stage stochastic setting with recourse.

[^1]
### 1.3.1 Individually-Fair Clustering

The most popular notion of individual fairness in algorithmic design was introduced in the seminal work of Dwork et al. [6], in the context of a classification problem. The high-level fairness idea established in [6] was that similar individuals should be treated similarly. For the classification problem studied in that paper, this meant that if two objects are similar according to some external metric, then the distributions over labels that they receive should be statistically close (the distribution over outcomes was naturally interpreted as the quality of service).

In order to transform the abstract concept of individual fairness introduced by Dwork et al. [6] into a rigorous and well-defined clustering problem, one needs to answer two questions:

1. How is similarity between points defined in a clustering problem?
2. What constitutes similar treatment in a clustering setting?

Obviously, different answers to the above two questions would yield different individually-fair clustering models. In what follows, we introduce two distinct clustering models that adhere to the fairness paradigm of Dwork et al. [6], with each of them answering the previous questions in its own unique manner.

### 1.3.1.1 Probabilistic Pairwise Fair Clustering

In this model, suppose that we are given a standard clustering input instance, which is augmented with a value $\psi_{j, j^{\prime}} \in[0,1]$ for every pair of points $j, j^{\prime} \in \mathcal{C}$. We assume, exactly in the same manner as the seminal work of Dwork et al. [6], that $\psi$ is the true metric indicating similarity between pairs of points. Specifically, the smaller $\psi_{j, j^{\prime}}$ is the more similar the two points
$j$ and $j^{\prime}$, with 0 indicating absolute identity and 1 complete dissimilarity.
Given the previous setup, a crucial question that needs to be answered is what can be interpreted as unfair treatment for two similar points? To answer that, notice that in certain applications assigning two points to different clusters implies that we are treating them differently. To clarify this statement, think of the following potential scenario. Suppose that the two points $j, j^{\prime}$ correspond to students, and the task at hand is assigning students to schools (schools will be represented by the locations $\mathcal{F}$ in the metric). Then, if $j$ gets assigned to a good school while $j^{\prime}$ does not, $j^{\prime}$ would arguably feel unfairly treated if it is significantly similar to $j$, i.e. $\psi_{j, j^{\prime}}$ is small. Hence, the motivation for introducing this model comes from applications where similar treatment is defined as placing similar points in the same cluster; in other words similar points gain utility from getting assigned to the same cluster.

However, the question about how to combat unfairness in such cases remains. Our proposed solution to this is that randomization can introduce fairness. Specifically, we would seek stochastic solutions that would cluster each pair $\left\{j, j^{\prime}\right\}$ apart with probability at most $\psi_{j, j^{\prime}}$. Therefore, the more similar two points are, the less likely it is that they will end up in different clusters, and thus the less likely it is that they will be treated in a different fashion.

The complete formal problem definition follows.

Definition 1.3.1 (Probabilistic Pairwise Fair Clustering - Pairfairclu). We are given a set of points $\mathcal{C}$ and a set of locations $\mathcal{F}$ in a metric space characterized by the distance function $d:(\mathcal{C} \cup \mathcal{F})^{2} \mapsto \mathbb{R}_{\geq 0}$. The input also includes a positive integer $k$. Furthermore, we are given a list of pairs of points $\mathcal{P}=\left\{e_{1}, \ldots, e_{m}\right\}$, where for every $q \in[m]$ we have $e_{q}=\left\{j_{q}, j_{q}^{\prime}\right\}$ and $j_{q}, j_{q}^{\prime} \in \mathcal{C}$. Finally, we are also given a list of values $\psi=\left\{\psi_{1}, \ldots, \psi_{m}\right\}$, with $\psi_{q} \in[0,1]$ being
interpreted as the similarity score for $\left\{j_{q}, j_{q}^{\prime}\right\}$. The goal is to find 1) a set $S \subseteq \mathcal{F}$ with $|S| \leq k$, and 2) a distribution $\mathcal{D}$ over assignments $\phi: \mathcal{C} \mapsto S$, such that:

- Sampling $\phi \sim \mathcal{D}$ can be done efficiently, i.e., in polynomial time.
- The Bounded Separation Probability (BSP) constraint is satisfied:

$$
\operatorname{Pr}_{\phi \sim \mathcal{D}}\left[\phi\left(j_{q}\right) \neq \phi\left(j_{q}^{\prime}\right)\right] \leq \psi_{q} \text { for every } e_{q} \in \mathcal{P}
$$

- Some standard clustering objective function is minimized. The problem is studied under each of the following objectives:

1. $k$-center: Find minimum $R$, such that $\operatorname{Pr}_{\phi \sim \mathcal{D}}[d(\phi(j), j) \leq R]=1$ for all $j \in \mathcal{C}$
2. $k$-median $(p=1), k$-means $(p=2)$ : $\operatorname{Minimize}\left(\sum_{j \in \mathcal{C}} \mathbb{E}_{\phi \sim \mathcal{D}}\left[d(\phi(j), j)^{p}\right]\right)^{1 / p}$.

At this point, it is interesting to notice that unlike classical problems in combinatorial optimization, in PAIRFAIRCLU the required solution is not a single deterministic solution, but rather a distribution over potential solutions. Finally, since PAIRFAIRCLu generalizes already known NP-hard problems (set all $\psi$ values to 0 in order to get the corresponding vanilla clustering problem), we immediately get that all versions of it are NP-hard as well.

### 1.3.1.2 Equitable Clustering

In this clustering model, we will once again follow the paradigm of Dwork et al. [6], with our goal being treating similar points similary. However, here we will define similarity between points and similar treatment in a different way than what we did for PairFairClu.

At first, we view the assignment distance $d(j, \phi(j))$ as the quality of service point $j$ receives, and hence this is the quantity that really matters for the point. As we mentioned earlier, in
classical clustering applications $d(j, \phi(j))$ would correspond to how similar $\phi(j)$ is to $j$, and in facility-location applications to the distance $j$ needs to travel in order to reach its service-provider. Therefore, the smaller $d(j, \phi(j))$ is, the happier the point $j$.

In scenarios where the points correspond to selfish agents, it is natural to assume that they will be mindful of the quality of service other points receive. Specifically, a point $j$ may feel that it is being handled unfairly by a solution $(S, \phi)$, if $d(j, \phi(j))$ is not close enough to the quality of service a group $\mathcal{S}_{j}$ of other points enjoys. In this context, the points of $\mathcal{S}_{j}$ are exactly those which $j$ perceives as similar to itself, hence it arguably believes that it should obtain similar treatment as them.

As a practical example that demonstrates the motivation behind the current model, consider the following application in an e-commerce site, where the points of $\mathcal{C}$ correspond to its users. In order to provide relevant recommendations, the website needs to choose a set $S$ of $k$ representative users, and then assign each point to one of those based on a mapping $\phi: \mathcal{C} \mapsto S$. The recommendations $j$ gets will be based on $\phi(j)$ 's profile, and in this case the quantity $d(j, \phi(j))$ corresponds to how representative $\phi(j)$ is for $j$, and hence how suitable $j$ 's recommendations are. A point $j$ may feel unfairly treated, if points that are similar to it (points $j^{\prime}$ with small $d\left(j, j^{\prime}\right)$ ) get better recommendations and consequently better service (see, e.g., the work of Datta et al. [7] for studies on similar users receiving different types of job recommendations).

The next definition will formalize the previously described abstract notion of fairness via two rigorous and related constraints, which we incorporate into the $k$-center problem. We focus on $k$-center due to its numerous practical applications, but mostly because of its theoretical simplicity, which allows us to explore in depth the intricacies and the combinatorial structure of this novel notion of individually-fair clustering.

Definition 1.3.2 ( $\alpha$-Equitable $k$-center - EQCENTER). We are given a set of points $\mathcal{C}$ in a metric space characterized by the distance function $d: \mathcal{C} \times \mathcal{C} \mapsto \mathbb{R}_{\geq 0}$. Moreover, the input includes a positive integer $k$ and a value $\alpha \geq 1$. Finally, for every point $j \in \mathcal{C}$ we have a similarity set $\mathcal{S}_{j} \subseteq \mathcal{C}$, denoting the group of points $j$ perceives as comparable to itself. We also define $R_{j}=\max _{j^{\prime} \in \mathcal{S}_{j}} d\left(j, j^{\prime}\right)$, and thus have $\mathcal{S}_{j} \subseteq\left\{j^{\prime} \in \mathcal{C} \mid d\left(j, j^{\prime}\right) \leq R_{j}\right\}$. The goal in our problems of interest is to choose a set $S \subseteq \mathcal{C}$ of at most $k$ centers, and then find an assignment $\phi: \mathcal{C} \mapsto S$, such that the $k$-center objective , i.e., $\max _{j \in \mathcal{C}} d(j, \phi(j))$, is minimized. Further, we use two different constraints to capture the notion of fairness we aim to study.

- Per-Point Fairness $(P P)$ : When we study the problem under this constraint, we want to make sure that for all $j \in \mathcal{C}$ with $\mathcal{S}_{j} \neq \emptyset$, we have:

$$
\begin{equation*}
d(j, \phi(j)) \leq \alpha \cdot \min _{j^{\prime} \in \mathcal{S}_{j}} d\left(j^{\prime}, \phi\left(j^{\prime}\right)\right) \tag{1.1}
\end{equation*}
$$

Here $j$ is satisfied if its quality of service is at most $\alpha$ times the "best" quality found in $\mathcal{S}_{j}$. Equivalently, in this case we should guarantee that for all $j \in \mathcal{C}$ we have:

$$
d(j, \phi(j)) \leq \alpha \cdot d\left(j^{\prime}, \phi\left(j^{\prime}\right)\right), \quad \forall j^{\prime} \in \mathcal{S}_{j}
$$

- Aggregate Fairness ( $A G$ ): Here for each $j \in \mathcal{C}$ with $\mathcal{S}_{j} \neq \emptyset$, we want to guarantee that:

$$
\begin{equation*}
d(j, \phi(j)) \leq \alpha \frac{\sum_{j^{\prime} \in \mathcal{S}_{j}} d\left(j^{\prime}, \phi\left(j^{\prime}\right)\right)}{\left|\mathcal{S}_{j}\right|} \tag{1.2}
\end{equation*}
$$

Hence, here $j$ feels fairly treated if $d(j, \phi(j))$ is at most $\alpha$ times the average quality of $\mathcal{S}_{j}$.

We call our problem $\alpha$-Equitable $k$-Center, and denote it by EQCENTER. Moreover, we consider it either under constraint (1.1) or under constraint (1.2). When we study it under constriant (1.1) we refer to it as EQCENTER-PP, and similarly when we use constraint (1.2) we denote it by EQCenter-AG.

Constraint (1.1) provides a stronger notion of fairness, in that each point $j$ cares explicitly about the assignment distance of every point $j^{\prime} \in \mathcal{S}_{j}$. On the other hand, constraint (1.2) is weaker, in the sense that under it the points compromise to merely comparing their quality of service to the average quality obtained by their similarity set. Due to this, a solution for (1.1) also constitutes a solution for (1.2), and hence for the same instance the optimal value of EQCENTERAG must be no larger than that of EQCENTER-PP. This observation reveals an intriguing tradeoff between how strict we want to be in our fairness constraints, and how much we care about the overall objective cost. This trade-off is further explored in later sections.

Finally, both variants are trivially NP-hard, since when $\mathcal{S}_{j}=\emptyset$ for all $j$, the fairness constraints become redundant, and the problems reduce to $k$-center, which is already known to be NP-hard [8].

Observation 1.3.3. A modeling advantage of this problem comes in the way we define similarity between points. A common criticism for similarity defined through the values $\psi$ as in [6], is that pinpointing an exact similarity value in $[0,1]$ for every pair of points is highly impractical [ $9,10,11]$. On the other hand, constructing the similarity sets $\mathcal{S}_{j}$ required in our model is much more straightforward and realistic.

### 1.3.2 Fair Graph-Cut Problems

We begin by describing an existing problem of finding cuts in a graph. Let $G=(V, E)$ be an undirected graph with vertex set $V$ and edge set $E$, where $n=|V|$ and every $e \in E$ has a cost $w_{e} \in \mathbb{R}_{\geq 0}$. In addition, we are given a designated "source" vertex $s \in V$. We are concerned with attempting to mitigate some sort of "disaster" that begins at $s$ and infectiously spreads through the network via the edges. This means that vertices $v \in V$ that are connected to $s$, i.e., vertices for which there exists an undirected $s-v$ path in $G$, are at some sort of risk or disadvantage.

A natural approach to mitigate such a spread is to remove edges from $G$, in an attempt to disconnect as many vertices of the graph from $s$ as possible. Specifically, if we remove a cut-set or simply cut $F \subseteq E$ from the graph, we denote by $\operatorname{prot}(V, E \backslash F, s)$ the set of vertices in $V$ that are no longer connected to $s$ in the residual graph $G_{F}=(V, E \backslash F)$, and hence are protected from the infectious process. At a high-level, the edge removal strategy contains the disastrous event within the set $V \backslash \operatorname{prot}(V, E \backslash F, s)$. Observe now that there is a clear trade-off between the cost $w(F)$ of the cut $F$ and $|\operatorname{prot}(V, E \backslash F, s)|$, i.e., the more edges we remove the more vertices we may be able to save.

The aforementioned trade-off naturally leads to the following optimization problem, which we call Size Bounded Minimum Capacity Cut or SB-MinCC for short. Given a graph $G$ with source vertex $s$ and a integer target value $T>0$, we want to compute a cut $F \subseteq E$ with the minimum possible cost $w(F)$, such that at least $T$ vertices of $V$ are saved in $G_{F}=(V, E \backslash F)$, i.e. $|\operatorname{prot}(V, E \backslash F, s)| \geq T$. This problem is NP-hard as shown by Hayrapetyan et al. [12]. The work of Svitkina and Tardos [13] gave a $O\left(\log ^{2} n\right)$-approximation algorithm for SB-MinCC, while Hayrapetyan et al. [12], Eubank et al. [14] gave constant factor bicriteria algorithms for it,
i.e. algorithms that provide solutions that come within a constant factor of the optimal cut cost, but at the same time might not save at least $T$ vertices.

Inspired by the recent interest revolving around algorithmic fairness, our goal here is to incorporate fairness ideas in SB-MINCC, and initiate the discussion of fairness requirements for problems of finding cuts in graphs. To the best of our knowledge, our work was the first to combine fairness with this family of problems.

The first notion of fairness that we consider is the widely used Demographic Fairness. The high-level idea behind this concept is that the set of elements that require "service" consists of various subsets-say demographic groups-and the solution should equally and fairly treat and represent each of these groups. In our case, if the vertices of the graph belong to different groups, we would like our solution to fairly separate vertices of each of them from s. In this way, we will avoid outcomes that completely ignore certain groups for the sake of minimizing the objective function; for example scenarios where certain groups are disproportionately protected while others are left completely exposed. Hence, we define the following problem.

Definition 1.3.4 (Demographically Fair Cut - DemFairCut). In addition to a graph $G=$ ( $V, E$ ) with weights $\left\{w_{e}\right\}_{e \in E}$ and the source $s \in V$, for some integer $\gamma \geq 1$ we are given sets $V_{1}, V_{2}, \ldots, V_{\gamma}$ and values $f_{1}, f_{2}, \ldots, f_{\gamma}$, such that $\forall h \in[\gamma]^{3}$ we have $V_{h} \subseteq V$ and $0 \leq$ $f_{h} \leq 1$. Note that each $v \in V$ may actually belong to multiple sets $V_{h}$. Letting $n_{h}=\left|V_{h}\right|$, the goal is to find a cut $F \subseteq E$ with the minimum possible $w(F)$, subject to the constraint that $\left|V_{h} \cap \operatorname{prot}(V, E \backslash F, s)\right| \geq f_{h} \cdot n_{h}$ for all $h \in[\gamma]$. In words, if each $V_{h}$ is interpreted as a demographic, we want the minimum cost cut under the condition that at least an $f_{h}$ fraction of the points in $V_{h}$ are disconnected from s (for all $h$ ).

[^2]Instantiating this definition with different values of $f_{h}$ allows us to model a variety of fairness scenarios. For example, setting $f_{h}=1 / 2$ would let us guarantee an equitable solution that protects at least half the vertices of each $V_{h}$. Alternatively, we can set $f_{h}$ to be a decreasing function of $n_{h} / n$, and thus yield a solution that focuses more on protecting smaller demographics. Moreover, notice that SB-MINCC is a special case of DEmFAIRCUT, where $\gamma=1$ (we only have one demographic group) and $f_{1}=\frac{T}{n}$. Hence, DemFairCut is NP-hard, since SB-MinCC is already known to be NP-hard.

The second notion of fairness we consider is called Probabilistic Individual Fairness, and was first introduced in the context of robust clustering [15, 16]. According to it, the final solution should not simply be just one deterministic solution, but rather a distribution $\mathcal{D}$ over feasible deterministic/static solutions. Then, considering each input element individually, the probability that it will get "good service" in a randomly drawn solution from this distribution, should be at most some given (fairness related) parameter. Obviously, sampling from this constructed distribution $\mathcal{D}$ must be achievable in polynomial time, and we call such distributions efficientlysampleable. Under this notion of fairness, we avoid outcomes that deterministically prevent satisfactory outcomes for certain individuals, e.g., a certain individual always getting poor quality service in the returned solutions.

Incorporating the above concept of fairness in SB-MINCC, implies that besides the global guarantee of saving at least $T$ vertices, we also need to provide a stochastic guarantee for each individual vertex, ensuring it that in the final solution it will be disconnected from $s$ with a certain probability. For instance, ensure that each vertex gets disconnected with probability at least $1 / 2$, and hence no specific vertex enjoys preferential treatment. The formal definition follows.

Definition 1.3.5 (Probabilistic Individually Fair Cut - IndFAIRCUT). In addition to a graph $G=$ ( $V, E$ ) with weights $\left\{w_{e}\right\}_{e \in E}$, a target $T \in \mathbb{N}_{\geq 0}$ and source $s \in V$, for each $v \in V \backslash\{s\}$ we are also given a value $p_{v} \in[0,1]$. The goal is to find an efficiently-sampleable distribution $\mathcal{D}$ over the cuts $\mathcal{F}(B)=\{F \subseteq E: w(F) \leq B \wedge|\operatorname{prot}(V, E \backslash F, s)| \geq T\}$, such that $\operatorname{Pr}_{F \sim \mathcal{D}}[v \in$ $\operatorname{prot}(V, E \backslash F, s)] \geq p_{v}$ for each $v \in V \backslash\{s\}$, and $B$ is the minimum possible.

SB-MinCC is also a special case of IndFairCuT; we can always set $p_{v}=0$ for all $v \in V \backslash\{s\}$ and make the stochastic constraints void. Hence, IndFAIRCUT is also NP-hard.

Observation 1.3.6. In both problems, we can assume that the disastrous event starts simultaneously from a set of vertices $S$, instead of just a single designated vertex. This assumption is without loss of generality, since $S$ can be merged into a single vertex $s$ (by retaining all edges between $S$ and $V \backslash S$ ), thus giving an equivalent formulation that matches ours.

### 1.3.2.1 Motivating Applications for Fair Cuts

Regarding demographic fairness, consider the following potential application. The vertices of the graph $V$ would correspond to geographic areas across the globe, and an edge $(u, v) \in E$ would denote whether or not there is underlying infrastructure, e.g., highways or airplane routes, that can transport people between areas $u$ and $v$. The disastrous event in this scenario is the spread of a disease in a global health crisis. If an area $u \in V$ is "infected", then it is natural to assume that neighboring areas (i.e., areas $v \in V$ with $(u, v) \in E$ ) can also get infected if we allow people to travel between $u$ and $v$. A central planner will clearly try to break a set of connections $F \subseteq E$ from the infrastructure graph, such that the total cost $w(F)$ of these actions will be as small as possible, while some guarantee on the number of protected areas $|\operatorname{prot}(V, E \backslash F, s)|$ is
also satisfied. The value $w(F)$ can be interpreted as the economic cost of the proposed strategy $F$, e.g., the lost revenue of airline companies resulting from cancelling flights.

In terms of fairness, we can think of the areas $V$ as coming from $\gamma$ different countries, with $V_{h}$ being the areas associated with country $h \in[\gamma]$. Then, a fair solution would not tolerate a discrepancy in how many areas are protected across different countries. For example, a fair approach would be to ensure that each country has at least half of its areas protected, since the less "infected" areas each country has, the more easily it can keep its local crisis under control.

As far as individual fairness is concerned, consider a computer network facing the spread of a computer virus. In this scenario, we want to minimize the cost of the connections removed, such that the infectious process is kept under control and thus a certain number of users $T$ does not get infected. However, each individual user of the network would arguably prefer to be in the set of protected vertices. Our notion of individual fairness as studied in IndFAIRCuT, will ensure this in a stochastic sense, by using appropriate values $p_{v}$.

### 1.3.3 A Computational Epidemiology Problem

With the COVID-19 pandemic and future such pandemics in mind, computational epidemiology, powered by AI and efficient algorithms, has emerged as a vital discipline. There are two major sources of uncertainty in typical applications of computational epidemiology: how the disease will unfold probabilistically (we may have a good model for this, but have limited control over such stochasticity), and models for contact between members of a population (social-contact networks). In the problem defined here, we take a rigorous stochastic-optimization approach to develop approximation algorithms for epidemic control under such sources of uncertainty.

Now we formally define our problem of interest, which we call MinInfEdge. The goal in this problem is to mitigate the stochastic spread of a disease in a social contact-network, via the application of a social-distancing strategy.

Suppose that we have an undirected graph $G=(V, E)$ representing a social-contact network; vertices represent people and edges represent social connections. In addition, we are given an edge weight $c_{e} \geq 0$ for every edge $e \in E$, indicating the cost of removing the social connection $e$ (not allowing interactions between the people corresponding to the endpoints of the edge). Let also $n=|V|$ and $m=|E|$.

We adopt the most widespread tool for modeling the spread of an epidemic in a socialcontact network, i.e., an SIR random model [17, 18]. In an SIR model of disease spread, each node of the graph is in one of the states $S$ (susceptible), I (infectious) or R (recovered). We also assume that the infection simultaneously starts at a subset $I_{0} \subseteq V$. An infectious vertex $v$ infects each susceptible neighbor $u$ once, independently with some known probability $p_{e} \in$ $[0,1]$, where $e=(u, v) \in E$. This is equivalent to the following percolation process [17, 18]: consider a random subgraph $G(\vec{p})=(V, E(\vec{p}))$ obtained from $G$ by retaining each edge $e \in E$ independently with probability $p_{e}$ (and thus removing each edge with probability $1-p_{e}$ ). In particular, the probability that a set $V_{i n f}$ of vertices is reachable from $I_{0}$ in $G(\vec{p})$ is precisely equal to the probability that the set $V_{i n f}$ becomes infected during the SIR process. We will sometimes abuse notation and let $G(\vec{p})$ also represent the distribution over subgraphs thus obtained. Without loss of generality, we can also assume that $I_{0}$ consists of a single vertex $s$, since adding a metavertex $s$ with probability 1 edges to all vertices in $I_{0}$ yields an equivalent setting. Finally, some of our results will only hold for a uniform probability setting in which $p_{e}=p$ for all $e \in E$; in this case we denote the random graph $G(\vec{p})$ by $G(p)$.

A social distancing strategy corresponds to the removal of a subset $F \subseteq E$ of edges; for such an $F$, we denote by $\inf (V, E \backslash F, s)$ the number of vertices that are in the same connected component as $s$ in the residual graph $G_{F}=(V, E \backslash F)$. For simplicity, we refer to $F$ as a cut or a cut-set, though it need not always induce a cut in the graph. Finally, the expected number of infected vertices in the percolation process is easily seen to be $\mathbb{E}_{G(\vec{p})}[\inf (V, E(\vec{p}) \backslash F, s)]$.

Given a budget $B$, the goal of MinInfEDGE is to choose a set $F \subseteq E$ such that:

1. $c(F)=\sum_{e \in F} c_{e} \leq B$, i.e., the total cost of the set $F$ of edges to be removed, is at most $B$.
2. $\mathbb{E}_{G(\vec{p})}[\inf (V, E(\vec{p}) \backslash F, s)]$ is minimized, i.e., the expected number of infections (nodes reachable from $s$ when we remove the edges in $F$ and conduct the disease percolation on the remaining graph) is minimized.

Remark: Observe that MinInfEdge can be viewed as a stochastic generalization of the cut problem SB-MinCC, where the randomness comes as uncertainty in the edge realization.
$(\alpha, \beta)$-approximation: As in [12, 19], we focus on bicriteria approximation algorithms. We say that a solution $F \subseteq E$ is an $(\alpha, \beta)$-approximation if $c(F) \leq \alpha B$, and $\mathbb{E}_{G(\vec{p})}[i n f(V, E(\vec{p}) \backslash$ $F, s)] \leq \beta \cdot \mathbb{E}_{G(\vec{p})}\left[\inf \left(V, E(\vec{p}) \backslash F^{*}, s\right)\right]$, where $F^{*}$ is an optimal solution for the given instance.

### 1.3.3.1 Random Graph Models for Social-Contact Networks

It is well-recognized that with the ever-growing importance of networks and network science, we need good random-graph models for predictive applications, simulations, testing of new algorithms etc.: see, e.g., [20, 21].

In our context of social-contact networks, the random-graph model of Chung and Lu [22] is particularly useful. In this model, we have a set of vertices $V$, and a weight $w_{v}$ for every node
$v \in V$ that denotes its expected degree in the graph; let $w_{\min }=\min _{v} w_{v}$ and $w_{\max }=\max _{v} w_{v}$. The edges $E$ of the graph are determined via the following random process. For every $u, v \in V$, the probability of having the $(u, v)$ edge in $E$ is

$$
q_{u, v}=\frac{w_{u} w_{v}}{\sum_{r \in V} w_{r}},
$$

where these edges are present independently and self-loops are allowed. A natural assumption here is $w_{\text {min }}=O(1)$. A common instantiation of this model is with a power law, in which $n_{i}$, the number of nodes of weight $i$, satisfies $n_{i}=\Theta\left(n / i^{\beta}\right)$, with $\beta>2$ being a model parameter. In our work, we are using the power law instantiation every time we consider this model.

For contact-networks, the random graphs captured by the Chung-Lu model are more realistic than those of the simple Erdős-Renyi model [19]. The reason for this is imposing a specified degree sequence that models the heavy-tailed nature of real-world degree distributions.

We refer to MinInfEdge when the graph $G=(V, E)$ is from the Chung-Lu model as MinInF-CL. The random process for constructing the graph $G=(V, E)$ in the Chung-Lu model should not be confused with the percolation process occurring on $G$ during the spread of the disease. In the case of MININF-CL, the reader can view the whole process as happening in two steps. At first, $G=(V, E)$ is chosen randomly according to the Chung-Lu model. Afterwards, the disease starts its diffusion in the chosen network according to the probability vector $\vec{p}$.

### 1.3.4 Two-Stage Stochastic Clustering

Stochastic optimization, first introduced in the work of Beale [23] and Dantzig [24], provides a way for modeling uncertainty in the realization of the input data. Here we are inter-
ested in a stochastic optimization paradigm known as the 2-stage recourse model [25]. This paradigm evolves in two stages. In the first, we do not yet know the input instance we need to work with, and we are only given access to a distribution $\mathcal{D}$ that describes possible realizations of future data. Given that knowledge, we take some stage-I actions and commit to an anticipatory part of the solution $x$, incurring some $\operatorname{cost} c(x)$. In the second stage, an input instance ("scenario") $A$ is sampled from the distribution $\mathcal{D}$, and we can take some stage-II recourse actions $y_{A}$ with cost $f_{A}\left(x, y_{A}\right)$. If $X$ is the set of stage-I actions and $Y$ the set of recourse actions, the goal is to find a solution $x^{\star} \in X$ to minimize $f(x)=c(x)+\mathbb{E}_{A \sim \mathcal{D}}\left[q_{A}(x)\right]$, where $q_{A}(x)=\min _{y \in Y}\left\{f_{A}(x, y) \mid(x, y)\right.$ is a valid solution for $\left.A\right\}$.

To complete the description of a two-stage stochastic problem, one needs to define how knowledge of the distribution $\mathcal{D}$ is represented during stage-I. There are three main models proposed in the literature for that, namely the black-box model [26, 27, 28, 29, 30], the polynomialscenarios model [31, 32, 33, 34], and the independent-activations one [32, 35, 36]. Among the three aforementioned models, the most general and expressive one is the black-box model. In our work here, we are interested only in the black-box and polynomial-scenarios models. We later present the necessary definitions in the context of the problems we study.

In what follows, we formally define the general skeleton for a number of 2-stage stochastic clustering problems that our work aims to solve.

We are given a set of clients $\mathcal{C}$ and a set of facilities $\mathcal{F}$, in a metric space characterized by a distance function $d$. We let $n=|\mathcal{C}|$ and $m=|\mathcal{F}|$. Our paradigm unfolds in two stages. In the first, each $i \in \mathcal{F}$ has a cost $c_{i}^{I}$, but at that time we do not know which clients from $\mathcal{C}$ will need service, and we only have a description of the distribution $\mathcal{D}$ that governs the arrivals of clients later on. In the second stage, a scenario $A \subseteq \mathcal{C}$ is realized with probability $p_{A}$ according to $\mathcal{D}$,
and now each $i \in \mathcal{F}$ has a cost $c_{i}^{A}$. The clients of the realized scenario are precisely those that will require service from the facilities of $\mathcal{F}$. Using only the description of $\mathcal{D}$, we can proactively open a set of facilities $F_{I}$ in stage-I. Subsequently, when a scenario $A$ arrives in stage-II, we can augment the already constructed solution by opening some additional facilities $F_{A}$.

The objective function we minimize is the maximum covering distance or radius. Let $d(j, S)=\min _{i \in S} d(i, j)$ for any $j \in \mathcal{C}$ and for any $S \subseteq \mathcal{F}$. We then ask for $F_{I}$ and $F_{A}$, such that $d\left(j, F_{I} \cup F_{A}\right) \leq R$ for every $A$ that materializes and all $j \in A$, for the minimum $R$ possible. Furthermore, the expected opening cost of the returned solution is required to be at most some given budget $B$, i.e.,

$$
\sum_{i \in F_{I}} c_{i}^{I}+\mathbb{E}_{A \sim \mathcal{D}}\left[\sum_{i \in F_{A}} c_{i}^{A}\right] \leq B
$$

We call this problem Two-Stage Stochastic Supplier or 2S-Sup for short.
Finally, we assume that for every $j \in \mathcal{C}$ we have $\operatorname{Pr}_{A \sim \mathcal{D}}[j \in A]>0$; note that if this is not the case, then the presence of $j$ in the input is completely redundant.

### 1.3.4.1 Additional Stage-I Constraints

Beyond the basic version of the problem, we also consider variants where there are additional hard constraints on the set of chosen stage-I facilities.

In Two-Stage Stochastic Matroid Supplier or 2S-MATSUP for short, the input also includes a matroid $\mathcal{M}=(\mathcal{F}, \mathcal{I})$, where $\mathcal{I} \subseteq 2^{\mathcal{F}}$ is the family of independent sets of $\mathcal{M}$. In this case, we additionally require $F_{I} \in \mathcal{I}$. In Two-Stage Stochastic Multi-knapsack Supplier or 2S-MuSup for short, $L$ additional knapsack constraints are imposed on $F_{I}$. Specifically, we are given budgets $W_{\ell} \geq 0$ and weights $f_{i}^{\ell} \geq 0$ for every $i \in \mathcal{F}$ and every integer $\ell \in[L]$, such that the stage-

I facilities should satisfy $\sum_{i \in F_{I}} f_{i}^{\ell} \leq W_{\ell}$ for every $\ell \in[L]$. We call a 2S-MUSUP instance discrete, if all $f_{i}^{\ell}$ are integers, and for such an instance we define parameter $\Lambda=\prod_{\ell=1}^{L} W_{\ell}$.

### 1.3.4.2 Modeling the Stage-I Distributional Knowledge

As mentioned earlier, to complete the description of a two-stage problem, one needs to define how knowledge of $\mathcal{D}$ is represented in stage-I. The most general representation is the black-box model, where we only have access to an oracle that can sample scenarios $A$ according to $\mathcal{D}$. In this model, every time a scenario $A$ is revealed, either through the oracle or through an actual data realization, we also learn the facility-cost vector $c^{A}$ associated with it. We also consider the more restricted polynomial-scenarios model, where all scenarios $A$, together with their occurrence probabilities $p_{A}$ and their corresponding cost vectors $c^{A}$, are explicitly provided.

We use the suffixes BB and Poly to distinguish these settings. For example, 2S-Sup-BB is the previously defined 2 S-SUP in the black-box model. In both settings, our algorithms must have runtime polynomial in $n, m$. For the polynomial-scenarios case, the runtime should also be polynomial in the number of explicitly provided scenarios.

### 1.3.4.3 Motivational Applications

To see a practical application for the problems defined in this section, consider healthcare resource allocation, when trying to mitigate a disease outbreak through the preventive placement of testing sites. Suppose that $\mathcal{F}$ corresponds to potential locations that can host a testing center (e.g., hospitals, private clinics, university labs), $\mathcal{C}$ to populations that can be affected by a possible disease outbreak, and each scenario $A \in \mathcal{D}$ to which populations suffer the outbreak. Since im-
mediate testing is of utmost importance, a central decision maker may prepare testing sites, such that under every scenario, each infected population has the closest possible access to a testing center. Assembling these sites in advance, i.e., in stage-I, has multiple benefits; for example, the necessary equipment and materials might be much cheaper and easier to obtain before the onset of the disease. Furthermore, the choice to minimize the maximum covering distance, as opposed to the opening cost, reflects a policy valuing societal welfare more than economic performance.

In addition, there may be further requirements for $F_{I}$, irrespective of the stage-II decisions, which cannot be directly reduced to the budget $B$. Such cases motivate our decision to study extra stage-I constraints. For instance, we might have a constraint on the total number of personnel we want to occupy prior to the outbreak of the disease, assuming that facility $i$ requires $f_{i}$ people to keep it operational during the waiting period; such a scenario implies an additional budget constraint on $F_{I}$ and is modeled by $2 \mathrm{~S}-\mathrm{MUSUP}$. Moreover, having a stage-I partitionmatroid constraint as in 2 S-MATSUP, can incorporate some notion of fairness in the decisions a planner makes. For instance, consider the case of multiple demographic groups, each preferring a different subset of facilities to be opened. In that situation, a partition matroid constraint will not allow any demographic's preferences to be over-represented in the set $F_{I}$ (no demographic will get preferential treatment).

### 1.4 Contributions and Outline

### 1.4.1 Contributions to PairFairclu

Chapter 2 contains our results on individually fair clustering, beginning with Section 2.1 and our results on PAIRFAIRCLU. Initially, in Section 2.1.1 we demonstrate our first ever result
on this problem, which involved a restricted setting introduced in [37].
 More importantly, in this paper we assume that the similarity metric $\psi$ has a specific structure, and that is for every $j, j^{\prime}$ we have $\psi_{j, j^{\prime}}=\frac{d\left(j, j^{\prime}\right)}{\kappa R}$. The value $\kappa>0$ is a user-specified parameter, while $R$ is set by the algorithm and it satisfies $R_{u n f}^{*} \leq R \leq 2 R_{u n f}^{*} ; R_{u n f}^{*}$ is the value of the optimal solution to the underlying $k$-center instance where no fairness constraints are imposed. Looking more closely to the previous definition of $\psi$, there is a certain cut-off value $\kappa R$, such that only points whose pairwise distance is smaller than this can be considered similar, and the closer the two points according to the metric space the more similar they are. Although this way of modeling similarity is not general enough, there is still some merit to it. Interpreting the provided metric space as a close approximation of the true similarity relation is a natural assumption for the vast majority of clustering applications, where distance already is correlated with similarity. Our main result from [37] follows.

Theorem 1.4.1 ([37]). When $\mathcal{C}=\mathcal{F}$, for any $\kappa>0$ there exists an algorithm which finds $S \subseteq \mathcal{C}$ with $|S| \leq k$ and an efficiently-sampleable distribution $\mathcal{D}$ over assignments $\phi: \mathcal{C} \mapsto S$ such that:

- $\operatorname{Pr}_{\phi \sim \mathcal{D}}\left[\phi(j) \neq \phi\left(j^{\prime}\right)\right] \leq \frac{d\left(j, j j^{\prime}\right)}{\kappa R}$ for every $j, j^{\prime}$.
- $\operatorname{Pr}_{\phi \sim \mathcal{D}}\left[\max _{j \in \mathcal{C}} d(j, \phi(j))=O(R \log k)\right] \geq 1-k^{1-\frac{1}{\kappa}}$.

Note that because $R \leq 2 R_{u n f}^{*}$, if we set $\kappa$ to be some small constant in $(0,1)$, the previous algorithm will give a solution that with high probability produces clusters of maximum radius $O\left(R_{u n f}^{*} \log k\right)$. The latter constitutes an $O(\log k)$-approximate solution for PAIRFAIRClU.

The idea behind the algorithm in the previous theorem is the following. Initially, we use a vanilla approach for $k$-center in order to find at most $k$ clusters. Then, we blow-up the radius
of each cluster according to an exponential distribution, and this guarantees the final result. We mention that this algorithm is very time efficient and thus can be used for large problem instances.

Moving on, in Section 2.1.2 we present some refined results on PairFairclu. Compared to our earlier work, these results capture all objective functions of interest ( $k$-center, $k$-median and $k$-means), provide constant factor approximations, cover the $\mathcal{C} \neq \mathcal{F}$ case, and most importantly work for arbitrary values $\psi$. Our main result follows.

Theorem 1.4.2 ([38]). Let $\tau^{*}$ be the optimal value of a PAIRFAIRCLU instance for any of the objective functions of interest, and $\rho$ be the best approximation ratio for the vanilla version of the problem where no fairness constraints are imposed (regular $k$-center, $k$-median, $k$-means). Our algorithm chooses set $S_{k}$ and constructs distribution $\mathcal{D}$ over assignments $\phi: \mathcal{C} \mapsto S_{k}$, such that:

- $\left|S_{k}\right| \leq k$ and sampling a $\phi \sim \mathcal{D}$ can be done in polynomial time
- $\operatorname{Pr}_{\phi \sim \mathcal{D}}\left[\phi\left(j_{q}\right) \neq \phi\left(j_{q}^{\prime}\right)\right] \leq 2 \psi_{q} \quad \forall e_{q} \in \mathcal{P}$
- The objective function cost is at most $(\rho+2) \tau^{*}$

Since for all $k$-center, $k$-median and $k$-means there exist small constant factor approximation algorithms, the approximation ratio we achieve in the above is also constant.

The previous result relies on a two-step process. At first, we solve the vanilla/unfair version of the problem using any known $\rho$-approximation algorithm, and this in return will give a set of centers $S_{k}$. Then, we set up a Linear Program (LP) based on $S_{k}$, which models a problem of assigning the points of $\mathcal{C}$ to the centers of $S_{k}$. After proving that the fractional solutions of this LP satisfy some important structural properties, we use a randomized rounding procedure from [39] to find an integral assignment of points to cluster centers (in other words a clustering). In the end, the desired distribution $\mathcal{D}$ results from the randomness of the rounding process.

A special case of PairFairclu that is of some interest and has been studied before in the literature $[40,41,42,43,44,45]$, occurs when all $\psi$ values are set to 0 . Then, our problem reduces to what is known as clustering with must-link constraints. In this setting, notice that the results of the previous theorem give true constant factor approximation algorithm for all $k$-center, $k$-median and $k$-means with must-link constraints (the 2 -violation is now irrelevant). For the $k$-median and the $k$-means objectives, these were the first algorithms with theoretical guarantees. Moreover, in Section 2.1.2 we also give a refined algorithm for $k$-center with must-link constraints, which achieves a better approximation ratio of 2 .

Finally, Section 2.1.3 contains experimental results from implementations of all our algorithms for PAIRFAIRClu.

### 1.4.2 Contributions to EQCEnTER

In Section 2.2 we present our results for EQCenter [46]. We begin by investigating the combinatorial structure of our newly introduced fairness constraints. At first, a question that naturally arises is for what values of the parameter $\alpha$ are our problems well-defined? We call a problem well-defined if it always admits a feasible solution $(S, \phi)$, i.e., $|S| \leq k$ and $\phi$ satisfies the corresponding fairness constraint for all $j$. Ideally, we would like our problems to admit feasible solutions for any possible value of $\alpha$. However, we give the following negative result which indicates that absolute equity is not always achievable.

Theorem 1.4.3. For both problems EQCENTER-PP and EQCENTER-AG, there exist instances of them with $\alpha<2$ that do not admit any feasible solution.

We then proceed to show that for $\alpha \geq 2$ there is always a feasible solution, thus settling the
vital question about the regime of $\alpha$ for which our problems are well-defined.

Theorem 1.4.4. For both problems EQCENTER-PP and EQCENTER-AG, every instance of them with $\alpha \geq 2$ always admits a feasible solution.

Given that $\alpha \geq 2$ is the range we should focus on, we proceed by studying another crucial concept, and that is the Price of Fairness (PoF) [47, 48]. This notion is just a measure of relative loss in system efficiency, when fairness constraints are introduced. Specifically, for a given instance of either EQCENTER-PP or EQCENTER-AG, PoF is defined as the value of the optimal solution to our fair problem, over the value of the optimal solution to the underlying $k$-center instance, where we drop the fairness constraints from the problem's requirements. In other words, $\mathrm{PoF}=($ optimal fair value $) /($ optimal unfair value $)$. In the vast majority of fair clustering problems it is known that there exist instances with unbounded PoF. In line with that, we show:

Theorem 1.4.5. Both EQCENTER-PP and EQCEntER-AG have instances with unbounded PoF.

All mentioned structural results are proven for $k \geq 2$. The $k=1$ case is trivial, since one can efficiently try each $j$ as a center, see if any yields a feasible solution, and also find the optimal one among the computed feasible solutions. On the other hand, even when $k=2$ and we only have $\binom{|\mathcal{C}|}{2}+|\mathcal{C}|$ center sets to check, the number of assignments for each set of size 2 is $2^{|\mathcal{C}|}$.

Moving forward, we provide an approximation algorithm that covers instances with $\alpha \geq 2$ for EQCENTER-PP and EQCENTER-AG. The main body of the algorithm remains the same for the two problems, with minor differences to capture the unique characteristics of each case. The algorithm can be viewed as operating in two phases; first chooses centers and then creates the assignment for the points. Our process of choosing centers constitutes an extension of a result by Khuller and Sussmann [49]. Our center-choosing procedure gives useful guarantees regarding
the distances between the chosen centers, a feature that is crucially exploited in the assignment phase of the algorithm, where we carefully construct the mapping $\phi$. Our result is:

Theorem 1.4.6. We are given instance of either EQCENTER-PP or EQCENTER-AG, with $\alpha \geq 2$. Let $R^{*}$ be the optimal value of this instance and let $R_{m}=\max _{j \in \mathcal{C}} R_{j}$. Our algorithm returns $a$ feasible solution $(S, \phi)$ to either problem, for which $\max _{j \in \mathcal{C}} d(j, \phi(j)) \leq 5 \max \left\{R^{*}, R_{m}\right\}$.

We strongly believe that in realistic applications involving selfish agents we would have $R_{m}=O\left(R^{*}\right)$, and hence our constructed solution will be a $O(1)$-approximate one. To see why this can be true, consider the following reasoning. At first, a very common assumption used throughout the fair clustering/classification literature [ $6,37,38,50,51$ ] is that the metric distance $d$ already captures a notion of resemblance. For instance, recall the use-case for EQCENTER mentioned in Section 1.3.1.2, where the goal was to build a recommendation system. There, two users that are close under $d$ have comparable profiles, and thus can be seen as similar. Consider now the optimal solution for the unfair problem on our instance $(\mathcal{C}, k)$, and let $R_{u n f}^{*}$ be its value. In this solution, the triangle inequality implies that a point $j$ will never be placed in the same cluster as some other $j^{\prime}$ with $d\left(j, j^{\prime}\right)>2 R_{u n f}^{*}$. Hence, the optimal unconstrained/unfair solution that can be thought of as an expert when it comes to determining similarity (it constructs the most intra-similar clusters when distance corresponds to similarity), does not deem the two points comparable enough to place them in the same cluster. Thus, the "advice" of the optimal unfair solution yields $R_{m} \leq 2 R_{u n f}^{*} \leq 2 R^{*}$ ( $R_{u n f}^{*} \leq 2 R^{*}$ since the unfair problem is less constrained).

Nonetheless, if a central planner is not sure whether or not $R_{m}=O\left(R^{*}\right)$, but still wants to enforce an $O(1)$-approximate outcome, there is an easy way to accomplish the latter. The planner can first compute a lower bound $R_{f}$ for $R_{u n f}^{*}$. There are various ways for achieving this, and the
simplest one is using the thresholding technique of Hochbaum and Shmoys [8]. Afterwards, the planner publishes a value $\lambda=O\left(R_{f}\right)$, and the points construct their similarity sets $\mathcal{S}_{j}$ based on it, i.e., they are only allowed to "envy" points within distance at most $\lambda$. Besides guaranteeing a $O(1)$-approximate solution, this strategy also enjoys explainability merits. By publishing an $O\left(R_{u n f}^{*}\right)$ value, the planner informs the agents that even under optimal conditions this is the best service they may receive. Hence, the points focus their attention on close neighbors that may end up becoming their assigned centers, and thus might get better service than them.

Continuing with the description of our results for EQCENTER, we also study the PoF behavior of our main algorithms. Specifically, when the similarity sets satisfy certain properties, we show that our algorithms enjoy a bounded PoF. If $R_{u n f}^{*}$ is the optimal value of the underlying $k$-center instance where no fairness constraints are imposed, we prove the following.

Theorem 1.4.7. If $R_{j} \leq R_{u n f}^{*}$ for all $j \in \mathcal{C}$, a modification to our algorithm yields a solution ( $S, \phi$ ), such that: (i) $|S| \leq 2 k$, (ii) both constraints (1.1) and (1.2) are simultaneously satisfied by $\phi$, and (iii) $\max _{j \in \mathcal{C}} d(j, \phi(j)) \leq 5 R_{u n f}^{*}$.

Theorem 1.4.8. When for all $j \in \mathcal{C}$ we have $\mathcal{S}_{j}=\left\{j^{\prime} \in \mathcal{C}: d\left(j, j^{\prime}\right) \leq R_{d}\right\}$ for some $R_{d}=\psi R_{\text {unf }}^{*}$ with $\psi=O(1)$, our algorithm for EQCENTER-AG provides a feasible solution with maximum radius at most $5 \max \left\{\psi R_{u n f}^{*}, R_{u n f}^{*}\right\}$.

Furthermore, we mention that all algorithms for EQCENTER that appear in Section 2.2 are purely combinatorial (e.g., do not require convex programming), and hence very efficient and easily implementable.

Another question we study is the assignment problem for EQCENTER-PP and EQCENTERAG. Specifically, if we are given the optimal set of centers $S^{*}$, can we find the corresponding
optimal assignment $\phi^{*}$ ? In a vanilla clustering setting this is trivial, since assigning points to their closest center is easily seen to yield the necessary results. However, as is the case in almost all literature on fair clustering, in the presence of fairness constraints like (1.1) or (1.2), such an assignment is not necessarily correct. This was actually among the first observations made in the seminal work of Chierichetti et al. [52], which initiated the research area of fair clustering. Therefore, since from a theoretical perspective the assignment problem is fundamental in a clustering setting and because in our case it appears highly non-trivial, we choose to address it in order to achieve a deeper understanding of the nature of our novel fairness constraints. In the end, we manage to show that with a slightly intricate iterative algorithm, we can indeed compute the optimal assignment $\phi^{*}$ in polynomial time; this result is presented in Section 2.2.3.

Finally, Section 2.2.4 shows experimental results from the implementation of our algorithms for EQCENTER. An important take-home message from those simulations is that the stronger per-point notion of fairness captured by constraint (1.1), in practice leads to solutions with more or less the same objective function values as the solutions produced for the weaker constraint (1.2). Hence, the use of constraint (1.1) is highly recommended.

### 1.4.3 Contributions to Fair Graph-Cut Problems

Chapter 3 contains our results for DEMFAIRCUT and InDFAIRCUT, which first appeared in [53]. In terms of our contribution here, we believe that the main aspect of it lies in introducing the first fair variants of graph-cut problems, thus initiating the discussion of fairness requirements in such computational settings.

In Section 3.1 we present a technique that is required in our approach for solving DEM-

Faircut and IndFairCut. The key insight is that we can reduce the problems on general graphs to the same problems on trees, by using a tree embedding from [54].

In Section 3.2 we address demographic fairness. At first, we provide an $O(\log n)$ approximation algorithm for DEMFAIRCUT based on dynamic programming. This algorithm runs in polynomial time only when the number of groups $\gamma$ is a constant. When $\gamma$ is not a constant and can be any arbitrary value, we develop a different algorithm based on a linear programming relaxation together with a dependent randomized rounding technique. This result yields an $O\left(\frac{\log n \log \gamma}{\epsilon^{2} \cdot \min _{h} f_{h}}\right)$-approximation for any $\epsilon>0$; however, the covering guarantee the LP-based algorithm provides to each demographic $V_{h}$ is only that at least $(1-\epsilon) f_{h} n_{h}$ vertices of it will be saved (in other words the algorithm is a pseudo-approximation). Regarding the dependence of the approximation ratio on $\min _{h} f_{h}$, we believe that in realistic fairness related applications the covering fractions $f_{h}$ should be relatively big, i.e., some constant $f_{h}=\Omega(1)$, since we care about protecting the vertices in the best way possible. Hence, the approximation ratio of our algorithm can be thought of as $O\left(\frac{\log n \log \gamma}{\epsilon^{2}}\right)$. Finally, we show that even on tree instances DEMFAIRCUT with arbitrary $\gamma$ is actually quite hard: it cannot be approximated better than $\Omega(\log \gamma)$. We do this by demonstrating an approximation factor preserving reduction from SET COVER.

In Section 3.3 we provide an $O(\log n)$-approximation algorithm for IndFaIRCuT. The high-level approach of this result relies on the round-or-cut framework developed by Anegg et al. [16], which we tailor in a way that suits the specific needs of our problem.

Finally, notice that since SB-MInCC is a special case of DEMFAIRCUT (with $\gamma=1$ ), and also a special case of IndFAIRCUT (when $p_{v}=0$ for all $v$ ), our dynamic programming algorithm from Section 3.2 and the algorithm of Section 3.3, both provide a $O(\log n)$-approximation for SB-MinCC. This constitutes an improvement over the best previously known $O\left(\log ^{2} n\right)$ -
approximation by Svitkina and Tardos [13].

### 1.4.4 Contributions to Computational Epidemiology

Chapter 4 contains all our results for MinInfEdge, which first appeared in [55]. We begin with Section 4.1, which demonstrates an $(O(1), O(1))$-approximation for unit edge-cost MinInfEdge (all edges of $G$ have cost 1 ). This result is for the uniform $p$ probability setting, in the regime where Karger's cut sparsification result holds [56]. Let $\hat{G}$ be a weighted graph obtained from $G$ by setting the weight $w_{e}$ of each edge $e \in E$ equal to $p$, and let $\hat{c}$ denote the weight of the minimum cut in $\hat{G}$. Karger's result states that if $\hat{c} \geq 9 \ln n$, then the size of every cut in $G(p)$ is close to the corresponding cut in $\hat{G}$. In this case, we are able to reduce MinInfEdge to a problem from [12], using just one random sample from $G(p)$. However, even this restricted setting is not trivial, this is actually the first rigorous result when the transmission probabilities in MinInfEdge are not 1 (the $p=1$ case corresponds to variants of SB-MinCC).

In Section 4.2 we present a sampling framework for MinInfEdge that utilizes the powerful sample-average-approximation (SAA) approach [57, 58, 59, 60]. Specifically, we sample a polynomial number of graphs from $G(\vec{p})$ and then formulate a linear program (LP) that describes the empirical estimate of the optimal solution on those samples. Afterwards, we solve this LP and provide a randomized-rounding procedure that transforms the fractional LP solution into an integral one. Let $F_{0}$ be the solution (set of edges to remove) that we compute, $O P T$ the value of the optimal solution, and $\Gamma$ the expected number of simple paths ${ }^{4}$ in a randomly drawn graph from $G(\vec{p})$, where the randomness also includes the random choice of $G$.

Three different sources of randomness/uncertainty: Our statements will refer to (com-

[^3]binations of) three distinct sources of randomness:

- Type 1: This randomness is over the random choice, if any, of our network $G=(V, E)$ (such as randomness resulting from choosing $G$ according to the Chung-Lu model). If the network $G$ is deterministic, Type 1 is vacuous: there is no randomness.
- Type 2: This randomness arises from the choices of our randomized rounding algorithm.
- Type 3: This type of randomness refers to the random percolation/diffusion of the disease in the given graph $G$, which is governed by $\vec{p}$.

Our main theorem for the SAA approach of Section 4.2 is summarized in the following, where "log" denotes the natural logarithm.

Theorem 1.4.9. For any chosen constants $\epsilon>0$ and $\gamma>1$, the following hold:

- with probability at least $1-O\left(n^{-\gamma}\right)$, where the randomness is solely of Type 2 , we have

$$
c\left(F_{0}\right) \leq O\left(\frac{\gamma}{\epsilon}\right) \log n \cdot B
$$

- there exists an event $\mathcal{A}$ with

$$
\operatorname{Pr}[\mathcal{A}] \geq 1-O\left(\frac{1}{n^{2}}\right)-O\left(\frac{\Gamma \log n}{\epsilon^{2} n^{\gamma}}\right)
$$

and

$$
\mathbb{E}\left[\inf \left(V, E(\vec{p}) \backslash F_{0}, s\right) \mid \mathcal{A}\right] \leq(1+\epsilon) \cdot O P T
$$

Here, randomness is with respect to Type 1 (if applicable), Type 2, and Type 3.

Observe now that if $\Gamma=\operatorname{poly}(n)$, we can choose $\gamma$ to be large enough, such that $\operatorname{Pr}[\mathcal{A}] \geq$ $1-O\left(1 / n^{2}\right)$. As we show in Section 4.2 this immediately implies the following corollary.

Corollary 1.4.10. When $\Gamma=\operatorname{poly}(n)$, we have

$$
\mathbb{E}\left[\inf \left(V, E(\vec{p}) \backslash F_{0}, s\right)\right] \leq(1+O(\epsilon)+O(1 / n)) O P T
$$

where the randomness is with respect to Type 1 (if applicable), Type 2, and Type 3.

Given the previous corollary, in Section 4.3 we prove that a family of Chung-Lu randomgraphs satisfies the $\Gamma=\operatorname{poly}(n)$ property (recall that this model captures realistic social-contact networks well [19, 61]). Under this property, our main result informally says that we can approximate the budget to within a factor $O(\log n)$ with high probability, and the expected number of infected people to within a constant factor. More specifically, Section 4.3 shows a phasetransition phenomenon as a function of the model parameter $\beta$ : for $\beta>3$ we get $\Gamma \leq \operatorname{poly}(n)$, while $\beta \leq 3$ implies that $\Gamma$ grows exponentially in $n$.

Finally, in Section 4.4 we show a slightly different SAA approach combined with a deterministic rounding which gives an $\left(O\left(n^{2 / 3}\right), O\left(n^{2 / 3}\right)\right)$-approximation for general graphs.

### 1.4.5 Contributions to Two-Stage Stochastic Clustering

In Chapter 5 we show our results on $2 \mathrm{~S}-\mathrm{SUP}$ and its related variants, which first appeared in [62]. In all these problems, our ultimate goal is to devise algorithms for the black-box setting, since the latter is the most general distributional model in the two-stage paradigm. As is usual in two-stage stochastic problems, we achieve our goal in three steps. First, we develop algorithms for the less complicated polynomial-scenarios model. Second, we sample a small
number of scenarios from the black-box oracle and use our polynomial-scenarios algorithms to (approximately) solve the problems on them. Finally, we extrapolate this solution to the original black-box problem. This overall methodology is called Sample Average Approximation (SAA).

Unfortunately, standard SAA approaches $[60,63]$ cannot be directly applied in radius minimization problems. On a high level, the obstacle here is that in our problems we need to compute the true cost of the approximate solution, something that is impossible using already existing results. Because this is a delicate technical issue, we refer the reader to Section 5.6 for an in-depth discussion. Therefore, we had to develop a novel sampling scheme.

Brief description of our sampling framework: Since the optimal black-box radius $R^{*}$ is always the distance between a client and a facility, there are at most $n m$ different options for it in any of our problems of interest. Thus, we consider each option separately, and assume for now that we work with a specific guess $R$. Given this, for some $N$ that is going to be defined later, we sample $N$ scenarios from the oracle, and let $Q=\left\{S_{1}, S_{2}, \ldots, S_{N}\right\}$ be that sampled set. We then run our polynomial-scenarios $\eta$-approximation algorithms on $Q$, which are guaranteed to provide solutions that cover each client within distance $\eta R$. Crucially, we show that if $R \geq R^{*}$ and $N$ is chosen appropriately, these solutions have cost at most $(1+\epsilon) B$ on $Q$, for any $\epsilon>0$. Hence, in the end we keep the minimum guess for $R$ whose cost over the samples is at most $(1+\epsilon) B$. For this minimum guess $R$ (which obviously satisfies $R \leq R^{*}$ ), the polynomial-scenarios algorithm returned a stage-I set $F_{I}$, and a stage-II set $F_{S_{v}}$ for each $S_{v} \in Q$. Our polynomial-scenarios algorithms are also designed to satisfy two additional key properties. First, given $F_{I}$ and any $A \notin Q$, there is an efficient process to extend the algorithm's output to a stage-II solution $F_{A}$ with $d\left(j, F_{I} \cup F_{A}\right) \leq \eta R$ for all $j \in A$. Second, irrespective of $Q$, the set $\mathcal{S}$ of possible black-box solutions the extension process might produce, has only exponential size as a function
of $n$ and $m$ (by default, $\mathcal{S}$ can have size $2^{m|\mathcal{D}|}$, and note that $\mathcal{D}$ may be exponentially large or even uncountably infinite). We call algorithms satisfying these properties efficiently generalizable . After using the extension process to construct a solution for every $A$ that materializes, there is a final scenario-discarding step to our framework. Specifically, for some given $\alpha \in(0,1)$, we first determine a threshold value $T$ corresponding to the $\lceil\alpha|Q|\rceil^{\text {th }}$ costliest scenario of $Q$. Then, if for an arriving $A$ the computed set $F_{A}$ has stage-II cost more than $T$, we perform no stage-II openings by setting $F_{A}=\emptyset$ (i.e., we "give up" on $A$ ). This step coupled with the bounds on $|\mathcal{S}|$ ensure that the overall opening cost of our solution is at most $(1+\epsilon) B$. At this point, note that discarding implies that there may exist scenarios $A$ with $d\left(j, F_{I} \cup F_{A}\right)>\eta R$ for some $j \in A$. However, we show such scenarios occur with probability at most $\alpha$, and the latter is user-specified and thus can be made inverse polynomially small.

In Section 5.2, we present our generalization scheme. We summarize it as follows:

Theorem 1.4.11. Suppose we have an efficiently generalizable, $\eta$-approximation for the polynomial scenarios variant of any of the problems we study. Let $\mathcal{S}$ be the set of all potential black-box solutions its extension process may produce. Then, with $\mathcal{O}\left(\frac{1}{\epsilon \alpha} \log \left(\frac{n m|\mathcal{S}|}{\gamma}\right) \log \left(\frac{n m}{\gamma}\right)\right)$ samples for any $\gamma, \epsilon, \alpha \in(0,1)$, we compute a radius $R$ and a black-box solution $F_{I}, F_{A}$ for all $A \in \mathcal{D}$ :

1. $F_{I}$ satisfies the stage-I specific constraints of the problem (matroid or multiknapsack).
2. $R \leq R^{*}$ and $\sum_{i \in F_{I}} c_{i}^{I}+\mathbb{E}_{A \sim \mathcal{D}}\left[\sum_{i \in F_{A}} c_{i}^{A}\right] \leq(1+\epsilon) B$, with probability at least $1-\gamma$. Here $R^{*}$ is the optimal radius of the black-box variant.
3. $\operatorname{Pr}_{A \sim \mathcal{D}}\left[d\left(j, F_{I} \cup F_{A}\right) \leq \eta R, \forall j \in A\right] \geq 1-\alpha$, with probability at least $1-\gamma$.

Theorem 1.4.12. We provide the following efficiently generalizable algorithms:

- A 3-approximation for 2S-SUP-Poly with $|\mathcal{S}| \leq(n+1)$ !.

For the black-box case, the sample complexity of Theorem 1.4.11 is $\tilde{O}\left(\frac{n}{\epsilon \alpha}\right)$.

- A 5-approximation for 2 S-MATSUP-Poly with $|\mathcal{S}| \leq 2^{m} n$ !. For the black-box case, the sample complexity of Theorem 1.4.11 is $\tilde{O}\left(\frac{m+n}{\epsilon \alpha}\right)$.
- A 5-approximation for discrete instances of 2S-MUSUP-Poly, with $|\mathcal{S}| \leq 2^{m}$ and runtime $\operatorname{poly}(n, m, \Lambda)$. In the black-box case, the sample complexity of Theorem 1.4.11 is $\tilde{O}\left(\frac{m}{\epsilon \alpha}\right)$.

Here, $\tilde{O}()$ hides polylog $(n, m, 1 / \gamma)$ terms. The 3 -approximation for 2 S-SUP-Poly is presented in Section 5.3. It relies on a novel LP rounding technique, not used in clustering problems before. Notably, its approximation ratio matches the lower bound of the non-stochastic counterpart [64] (Knapsack Supplier) of 2S-SuP, something very rare in the two-stage paradigm. The 5-approximation for 2S-MATSUP-Poly is presented in Section 5.4. It relies on solving an auxiliary LP, whose optimal solution is guaranteed to be integral. The 5 -approximation for 2 S -MUSUP-Poly is presented in Section 5.5, and is based on a reduction to a deterministic supplier problem with outliers. Specifically, if we view stage-I as consisting of a deterministic robust problem, stage-II is interpreted as trying to cover all outliers left over by stage-I.

### 1.4.5.1 Further Comments On Our Generalization Scheme

The main advantages of our generalization scheme are:

1. Unlike standard SAA approaches Swamy and Shmoys [60], Charikar et al. [63], it can handle problems based on the maximum-radius objective function.
2. The approximation ratio $\eta$ is preserved with high probability during the generalization. By contrast, in typical two-stage problems, the approximation ratio usually gets inflated when generalizing the polynomial-scenarios setting to the black-box one.
3. The adaptive selection of $T$ yields crisp bounds in terms of $\alpha, \epsilon$. By contrast, simpler nonadaptive approaches (e.g., $T=\frac{B}{\alpha}$ ) would still give the same guarantees, but the dependence of the sample bounds on $\alpha, \epsilon$ would be worse ( $\frac{1}{\epsilon^{2} \alpha^{2}}$ compared to $\frac{1}{\epsilon \alpha}$ as we achieve). This adaptive thresholding may also be of independent interest; we conjecture that it might be able to improve the sample complexity in the SAA analysis of Charikar et al. [63].

Remark: There is an important connection between the design of our generalization scheme and the design of our polynomial-scenarios approximation algorithms. In any SAA approach, the sample complexity necessarily depends on the set of possible actions over which the generalization is performed. In Theorem 1.4.11, the sample bounds are given in terms of the cardinality of $\mathcal{S}$. Following the lines of Swamy and Shmoys [60], it may be possible to replace this dependence with a notion of dimension of the underlying convex program. However, such general bounds would lead to significantly larger complexities, consisting of very high order polynomials of $n, m$. On the other hand, all of our polynomial-scenarios algorithms are carefully designed, so that the cardinality of $\mathcal{S}$ itself is small. Indeed, one of the major contributions of this work is to show that this property can still be satisfied for sophisticated approximation algorithms using complex $L P$ rounding. Consequently, we can use simple generalization bounds. Besides being clear and intuitive, these lead to a much lower dependence on $n, m$ for the sample complexity. To our knowledge, these are the first examples of non-trivial algorithms for two-stage stochastic problems via directly bounding the size of the solution set $\mathcal{S}$.

### 1.5 Literature Review

### 1.5.1 Fair Clustering Literature

Fair clustering has been a very active and prolific area of research in the last few years, with a plethora of fairness notions studied. We next demonstrate a rigorous taxonomy of all concepts of fairness in clustering, together with all related results.

### 1.5.1.1 Demographic Fairness Literature

The most popular and widely-studied notions of fairness in clustering revolve around Demographic Fairness. The high-level goal of such notions is to treat groups of points fairly, with respect to either how other groups are being treated or with respect to the specific needs of the group at hand. Usually, the groups of points that we are interested in are given as part of the input, and they represent points sharing some important defining attributes. Hence, each group can be perceived as an underlying demographic of the population. For example, we might be provided with groups representing age, e.g., "children", "teenagers", "adults", "senior citizens". We now present a detailed taxonomy of all concepts of demographic fairness in clustering.

Balanced Representation: This is the first fair clustering problem ever studied, and it was introduced in the seminal work of Chierichetti et al. [52]. In that paper, the authors study a clustering problem where the points of the metric space are partitioned into two demographics, say red and blue points, based on some protected attribute. What is then required, is a solution in which the fraction of red and blue points in every cluster is identical to the global fraction of the corresponding color class across the whole dataset. Chierichetti et al. [52] study this
problem under the $k$-center and $k$-median objectives, and provide constant factor approximation algorithms. Huang et al. [65], Backurs et al. [66] proceeded to give scalable algorithms for this problem. Further, this framework was later generalized to multiple demographics for the $k$-center objective by Bercea et al. [67], Rösner and Schmidt [68], as well as to other clustering paradigms such as spectral clustering [69] and correlation clustering [70].

Relaxed Notion of Balance: Here we do not require the ratio of the points of each demographic in the cluster to be exactly the same as the ratio of its points over the whole dataset. We rather relax this condition, and ask for a ratio that is within a certain input given range. This concept was simultaneously introduced by Bercea et al. [67], Bera et al. [71], Ahmadian et al. [72] in the context of $k$-clustering (for all $k$-center, $k$-median, $k$-means objectives). This relaxed notion allowed for small constant factor approximations for all clustering objectives. In addition, the model of Bera et al. [71] also captures the intriguing case of overlapping demographics. Finally, this concept was later studied for hierarchical clustering by Ahmadian et al. [73], in a stochastic setting by Esmaeili et al. [74], and with a fairness maximization objective [75].

Fairness in Center Selection: In the context of $k$-clustering where $k$ cluster centers need to be selected, it is reasonable to impose fairness constraints on the set of selected centers. The work of Kleindessner et al. [76], Jones et al. [77] focuses on avoiding over-representation of any demographic group in the set of selected centers, while Thejaswi et al. [78] focus on avoiding under-representation of any group in the chosen centers.

Proportionally Fair Clustering: In this model, we assume that every large enough group of points is entitled to their own cluster center. Therefore, we should make sure that for every such group of points, their assigned centers are close enough to all of them, in the sense that there exists no other point that is closer to every point in the coalition, thus giving them an incentive
to deviate from the current assignment. An interesting aspect of this model is that there are no a priori given sets of points, and the fairness guarantee should simultaneously hold for every possible subset of the dataset. This setting was introduced by Chen et al. [79], with further refined results given by Micha and Shah [80].

Demographic Fairness in Clustering with Outliers: When clustering with outliers, we are allowed to exclude a certain number of points from our computations, thus leaving them "unclustered". Being chosen as an outlier is an inherently disadvantageous event for a point. Hence, when the dataset involves points coming from different demographic groups, we should make sure that a fair amount of points from each such group is chosen as outliers. This model has been studied only under the $k$-center objective, with [81] being the paper that introduced it. Later on, Anegg et al. [16], Jia et al. [82] gave improved results for it.

Socially Fair $k$-Clustering: In many clustering applications, the quantity that really matters to each point is the distance to its assigned center. Hence, in the presence of multiple demographic groups it makes sense to consider a fairness metric that looks at the average assignment distance of each demographic. Abbasi et al. [83], Ghadiri et al. [84] independently introduced a model where the objective is to minimize the maximum average assignment distance over all demographic groups. Later on, Makarychev and Vakilian [85], Goyal and Jaiswal [86], Ghadiri et al. [87] gave improved results for this problem.

### 1.5.1.2 Individual Fairness Literature

The high-level goal of notions of individuals fairness is to treat each individual point fairly, with respect to either how other individual points are being treated or with respect to the specific
needs of the point at hand. Therefore, our attention here shifts from groups to individuals. The seminal work of Dwork et al. [6] will help us create a taxonomy for concepts of individual fairness in clustering. Specifically, Dwork et al. [6] introduced a notion of individual fairness for classification, where the aim was to treat similar individuals similarly. Our categorization makes a distinction between notions of fairness for clustering that follow the paradigm of Dwork et al. [6], and notions that do not.

Besides our work in [37, 38, 46], clustering papers on individually-fair clustering that adhere to the notion of Dwork et al. [6] include Anderson et al. [50], Kar et al. [88].

Anderson et al. [50] study a similar setting as the one of PairFairClu. Specifically, they are looking for a set of centers $S$, but for all $j \in \mathcal{C}$ they require a distribution $\phi_{j}$ that assigns $j$ to each $i \in S$ with probability $\phi_{i, j}$. Given that, they seek solutions that minimize the clustering objectives, while ensuring that for given pairs $j, j^{\prime}$, their assignment distributions are statistically similar based on some metric $D$ that captures distributional proximity (e.g., total variation and KL-divergence). In other words, they interpret individual fairness as guaranteeing $D\left(\phi_{j}, \phi_{j^{\prime}}\right) \leq \psi_{j, j^{\prime}}$ for all provided pairs $\left\{j, j^{\prime}\right\}$ with similarity value $\psi_{j, j^{\prime}}$. Although this work is certainly interesting, it has a significant modeling issue. To be more precise, suppose that for $j, j^{\prime}$ the computed $\phi_{j}, \phi_{j^{\prime}}$ are both the uniform distribution over $S$. Then, according to Anderson et al. [50] a fair solution is achieved. However, the actual probability of placing $j, j^{\prime}$ in different clusters (hence treating them unequally) is almost 1 (if there is no correlation between $\phi_{j}$ and $\left.\phi_{j^{\prime}}\right)$. On the other hand, our definition of PairFairClu which instead asks for a distribution $\mathcal{D}$ over assignments $\phi: \mathcal{C} \mapsto S$, always provides meaningful results, since it bounds the quantity that really matters, i.e., the probability of separating $j$ and $j^{\prime}$ in a random $\phi \sim \mathcal{D}$.

In the work of Kar et al. [88] that also follows the paradigm of Dwork et al. [6], each point
$j$ has a set $\mathcal{S}_{j} \subseteq \mathcal{C}$ of other points that are similar to it, as well as a tolerance parameter $m_{j}$. The goal is then to find solutions minimizing some clustering objective, while ensuring that each $j \in \mathcal{C}$ will have at least $m_{j}$ points from $\mathcal{S}_{j}$ in the cluster it is assigned to.

Next we provide a taxonomy for notions of Individual Fairness in clustering that diverge from the paradigm of Dwork et al. [6].

A Center in My Neighborhood: In certain applications, minimizing a global objective on the assignment distances, e.g., the $k$-means objective, does not suffice in order to capture the special needs of individual points. For instance, some points may require an assignment distance much smaller than what other points can tolerate. Hence, it is reasonable to seek solutions where we do not only minimize the global objective, but also try to satisfy the individual distance needs of every point. This model was introduced in [51], and increasingly better results were given in a subsequent series of papers [89, 90, 91].

Individual Fairness when clustering with outliers: As discussed earlier, being chosen as an outlier is an inherently disadvantageous event for a point. Therefore, a solution that consistently picks certain points as outliers can be arguably considered as biased against them. Harris et al. [15] introduced a randomized model, where each point should not be picked as an outlier with probability at least a certain value. Hence, in a stochastic sense this type of solution guarantees that every individual point has a decent chance of getting clustered. Later on, Anegg et al. [16] gave better results for this problem. This stochastic notion of individual fairness is exactly the one we study in IndFairCut.

Proximity to the points of your cluster: Finally, Kleindessner et al. [92], Ahmadi et al. [93] introduce an intriguing concept, that views individual fairness as ensuring that each point is on average closer to the points in its own cluster than to the points in any other cluster.

### 1.5.2 Stochastic Clustering Literature

Regarding clustering problems in the two-stage stochastic model with recourse, most prior work has focused on Facility Location [26, 28, 29, 30, 31, 33, 94]. On similar lines, [95] studies a stochastic $k$-center variant, where points arrive independently, but each point only needs to get clustered with some given probability. Furthermore, the only distributional model that we do not consider in our work is the independent activations one. According to it, each input element, e.g., in our case the clients, arrives independently with some known probability.

Finally, Harris et al. [96, 97] consider versions of classical clustering problems, where besides the overall objective function minimization, we are also required to provide a stochastic solution with per-point distance guarantees on expectation.

### 1.5.3 Vanilla Clustering Literature

Here we mention the best known results for the "unfair" variants of all clustering problems that this work studies. For the classical $k$-center problem the best known approximation ratio is 2 , and there are two different algorithms achieving it [8, 98]. Moreover, this ratio of 2 is the best-possible unless $\mathrm{P}=\mathrm{NP}$ [64]. For the $k$-median problem, the best known approximation ratio is 2.765 [99], while the best hardness is $1+2 / e$ [100]. Finally, Ahmadian et al. [101] give a 6.357-approximation for $k$-means, but the best hardness result here is marginally above 1 [102].

### 1.5.4 Graph-Cuts and Computational Epidemiology Literature

The unfair variant of our graph-cut problems, i.e., SB-MinCC, was studied in [12, 13, 14]. These papers also considered additional versions of SB-MINCC, where the goal was to maximize
$|\operatorname{prot}(V, E \backslash F, s)|$ (equivalently minimize $|V \backslash \operatorname{prot}(V, E \backslash F, s)|$ ) subject to an upper bound constraint on $w(F)$; note that the latter is exactly a deterministic variant of MinInfEdge, where all transmission probabilities are 1 (the case of a highly infectious disease).

As far as MinInfEdgE is concerned, there has been much work on heuristics for interventions in the SIR model [20, 103, 104, 105, 106, 107]. In particular, heuristics based on degree or centrality, e.g., $[105,106]$, have been shown to be quite effective in many classes of networks (including random graphs), but these do not provide any theoretical guarantees. The work of Sambaturu et al. [107] also explores the use of the sample average approximation method, but unlike our results it has worst-case approximation bounds as large as $O(n)$. We note that another related direction of work has been on reducing the first eigenvalue, referred to as the spectral radius, based on a characterization of the time to die out in SIS models (in which, unlike the SIR model, an infected node switches back to state S) [108]. There has been much work on reducing the spectral radius, e.g., $[109,110,111,112,113]$. However, these results do not imply any guaranteed bounds for MinInfEdge.

### 1.6 Pointers to the Most Important Results of the Thesis

Our best results for PairFairclu are presented in Section 2.1.2. For EQCenter the important Sections are 2.2.1 and 2.2.2. The former contains the structural properties of the problem, while the latter our algorithms. Our main result for DemFairCut is presented in Section 3.2.2 and our main result for IndFairCut in Section 3.3. Regarding MinInfEdge, the SAA framework is demonstrated in 4.2, with an application to Chung-Lu graphs shown in Section 4.3. Finally, the gist of our approach for two-stage clustering is contained in Sections 5.2, 5.3.

## Chapter 2: Addressing Individually Fair Clustering

### 2.1 Solving PairFairClu

### 2.1.1 Some First Simple Results

In this section we present our first ever result on PairFairClu, which appeared in [37]. For this result we are only going to focus on the $k$-center objective function. Recall now that in this setting the input consists of a set of clients $\mathcal{C}$, together with a distance function $d: \mathcal{C}^{2} \mapsto \mathbb{R}_{\geq 0}$.

The idea is to first run an "unfair" $k$-center algorithm and then order its returned clusters arbitrarily. Afterwards, we process each cluster sequentially according to the chosen order, and we expand its radius by a value sampled independently from an exponential distribution. Any point which falls within the radii of more than one of these expanded clusters is assigned to the earliest one in the ordering.

We use $C_{i} \subseteq \mathcal{C}$ to refer to the $i^{\text {th }}$ cluster found by the initial "unfair" algorithm and $c_{i}$ to refer to its center. Similarly, we use $C_{i}^{\prime}$ to refer to the $i^{\text {th }}$ expanded cluster that will be part of our fair output and $c_{i}^{\prime}$ to refer to its center. For readability, we also refer to $C_{i}$ and $c_{i}$ as original and $C_{i}^{\prime}$ and $c_{i}^{\prime}$ as final. Let $R_{i}=\max _{j \in C_{i}} d\left(c_{i}, j\right)$ be the radius of $C_{i}$ and $R=\max _{i} R_{i}$ be the maximum radius of any cluster found by the original clustering step. Let $\kappa$ be a user-specified constant greater than 0 . The approach is summarized in Algorithm 1.

```
Algorithm 1: Initial result for PAIRFAIRCLU
    Step 1: Run any vanilla \(k\)-center algorithm and order its returned clusters arbitrarily
        from 1 to \(k\). Let \(R\) be the maximum distance of any point to its center;
    Step 2: Let \(C_{i}\) be a set of points denoting cluster \(i\). Let \(c_{i} \in C_{i}\) be the center of \(C_{i}\) and
        \(R_{i}\) be the radius of \(C_{i}\);
    Step 3: Treat all points including centers as "unclustered" and construct a new set of
        clusters denoted by \(C_{i}^{\prime}\) as follows:
    for \(i=1 \boldsymbol{t o} k\) do
        4: Sample an independent random variable \(x_{i}\) from an exponential distribution
        with parameter \(\lambda=1 /(\kappa R)\). Let \(X_{i}\) be the realization of that random variable;
        5: Construct cluster \(C_{i}^{\prime}\) by adding every unclustered point within radius \(R_{i}+X_{i}\)
            from original center \(c_{i}\);
        6: If \(c_{i}\) was unclustered at the start of this iteration designate it as the center \(c_{i}^{\prime}\) of
        \(C_{i}^{\prime}\). Otherwise, if \(c_{i}\) has been added to a previous cluster \(C_{i^{\prime}}, i^{\prime}<i\), then choose
        any other previously unclustered point in \(C_{i}^{\prime}\) to be the center \(c_{i}^{\prime}\). If no such point
        exists, call the cluster empty;
    end
```

We note that in the for loop of steps 4 to 6 of Algorithm 1, the centers 1 through $k$ are processed in an arbitrary order. Because of this, our proofs also hold if the center are processed in a random order or some particular order aligned with another side objective.

We first prove that Algorithm 1 satisfies the BSP constraint for every pair of points $j, j^{\prime}$, with $\psi_{j, j^{\prime}}=d\left(j, j^{\prime}\right) /(\kappa R)$. At a high level, it is the memoryless property of the exponentially distributed random variables that allows our algorithm to achieve the guarantee in Lemma 2.1.1.

Lemma 2.1.1. For any pair of points $j$ and $j^{\prime}$, the probability that Algorithm 1 separates $j$ and $j^{\prime}$ into two separate clusters is at most $d\left(j, j^{\prime}\right) /(\kappa R)$, where $R$ is the maximum radius obtained by the initial algorithm used in step 1 and $\kappa>0$ is a user-specified constant.

Proof. For an arbitrary pair of points $j, j^{\prime} \in \mathcal{C}$, consider the first iteration $i$ in which at least one of the points is added to a final cluster $C_{i}^{\prime}$. Without loss of generality, let $j$ be the closer point to the original center $c_{i}$. If $d\left(c_{i}, j^{\prime}\right)<R_{i}$, both points will be added to $C_{i}^{\prime}$ regardless of the value of $X_{i}$ and the probability of separating $j$ and $j^{\prime}$ is 0 . Otherwise, conditioned on having
$R_{i}+X_{i}>d\left(c_{i}, j\right)\left(j\right.$ gets in $\left.C_{i}^{\prime}\right)$, the probability of separating them is the probability that the value $R_{i}+X_{i}$ will be smaller than $d\left(c_{i}, j^{\prime}\right)$ (otherwise $j^{\prime}$ will also get in $C_{i}^{\prime}$ hence there will be no separation). Therefore, we finally get:

$$
\begin{aligned}
\mathcal{P} & :=\operatorname{Pr}\left[j \text { and } j^{\prime} \text { are separated by } C_{i}^{\prime} \mid R_{i}+X_{i}>d\left(c_{i}, j\right)\right] \\
& =\operatorname{Pr}\left[R_{i}+X_{i} \leq d\left(c_{i}, j^{\prime}\right) \mid R_{i}+X_{i}>d\left(c_{i}, j\right)\right] \\
& =1-\operatorname{Pr}\left[R_{i}+X_{i}>d\left(c_{i}, j^{\prime}\right) \mid R_{i}+X_{i}>d\left(c_{i}, j\right)\right] \\
& \leq 1-\operatorname{Pr}\left[R_{i}+X_{i}>d\left(c_{i}, j\right)+d\left(j, j^{\prime}\right) \mid R_{i}+X_{i}>d\left(c_{i}, j\right)\right] \\
& =1-e^{-\lambda d\left(j, j^{\prime}\right)}=1-e^{-d\left(j, j^{\prime}\right) / \kappa R} \leq \frac{d\left(j, j^{\prime}\right)}{\kappa R}
\end{aligned}
$$

The fourth line follows by the triangle inequality which gives $d\left(c_{i}, j^{\prime}\right) \leq d\left(c_{i}, j\right)+d\left(j, j^{\prime}\right)$, and hence $\operatorname{Pr}\left[R_{i}+X_{i}>d\left(c_{i}, j\right)+d\left(j, j^{\prime}\right) \mid R_{i}+X_{i}>d\left(c_{i}, j\right)\right] \leq \operatorname{Pr}\left[R_{i}+X_{i}>d\left(c_{i}, j^{\prime}\right) \mid R_{i}+X_{i}>\right.$ $\left.d\left(c_{i}, j\right)\right]$. The rest of the calculations follow from properties of the exponential distribution.

We now bound the amount that the radius of any cluster will increase beyond the maximum value $R$ achieved by the original "unfair" algorithm from step 1 of Algorithm 1. At this point note that any 2-approximation algorithm for the classical $k$-center would guarantee $R \leq 2 R_{u n f}^{*}$, where $R_{u n f}^{*}$ is the optimal "unfair" radius.

Lemma 2.1.2. The maximum radius of any cluster is $O(R \log k)$ with high probability.

Proof. Since every $C_{i}^{\prime}$ has radius at most $X_{i}+R$, we will upper bound the probability that there exists a very large $X_{i}$. Via a union bounds and the properties of the exponential distribution:

$$
\operatorname{Pr}\left[\exists X_{i}>R \log k\right] \leq k \operatorname{Pr}\left[X_{i}>R \log k\right]=k e^{-\lambda R \log k}=k e^{-\log k / \kappa}=k^{1-1 / \kappa}
$$

### 2.1.2 More Refined Results for PairFairclu

Here we present some refined results for PairFairclu, which can be found in [38]. Before we proceed to our algorithms, we give some auxiliary lemmas.

### 2.1.2.1 An LP-Rounding Subroutine

We present an important subroutine developed by Kleinberg and Tardos [39], which we repeatedly use in our results, and call it KT-Round. Suppose we have a set of elements $V$, a set of labels $L$, and a set of pairs $E \subseteq\binom{V}{2}$. Consider the following Linear Program (LP).

$$
\begin{array}{lr}
\sum_{l \in L} x_{l, v}=1, & \forall v \in V \\
z_{e, l} \geq x_{l, v}-x_{l, w}, & \forall e=\{v, w\} \in E, \forall l \in L \\
z_{e, l} \geq x_{l, w}-x_{l, v}, & \forall e=\{v, w\} \in E, \forall l \in L \\
z_{e}=\frac{1}{2} \sum_{l \in L} z_{e, l}, & \forall e=\{v, w\} \in E \\
0 \leq x_{l, v}, z_{e}, z_{e, l} \leq 1, & \forall v \in V, \forall e \in E, \forall l \in L \tag{2.5}
\end{array}
$$

Theorem 2.1.3. [39] Given a feasible solution $(x, z)$ of (2.1)-(2.5), there exists a randomized rounding approach KT-Round $(V, L, E, x, z)$, which in polynomial expected time assigns each $v \in V$ to a $\phi(v) \in L$, such that:

1. $\operatorname{Pr}[\phi(v) \neq \phi(w)] \leq 2 z_{e}, \quad \forall e=\{v, w\} \in E$
2. $\operatorname{Pr}[\phi(v)=l]=x_{l, v}, \quad \forall v \in V, \forall l \in L$

### 2.1.2.2 Our Main Algorithmic Result

We use $k$-center-BSP, $k$-median-BSP, $k$-means-BSP to denote PAIRFAIRCLU under the $k$ center, $k$-median and $k$-means objectives respectively. In this section we show how to achieve approximation algorithms with provable guarantees for all problems of interest, using a general two-step framework. At first, let $P_{k}$ denote any of the vanilla versions of the objective functions we consider, i.e., $P_{k} \in\{k$-center, $k$-median, $k$-means $\}$.

To tackle a $P_{k}$-BSP instance, we begin by using on it any known $\rho$-approximation algorithm $A_{P_{k}}$ for $P_{k}$. This gives a set of locations $S_{P_{k}}$ and an assignment $\phi_{P_{k}}$, which yield an objective function cost of $\tau_{P_{k}}$ for the corresponding $P_{k}$ instance. In other words, we drop the BSP constraints from the $P_{k}$-BSP instance, and simply treat the problem as its vanilla counterpart. Although $\phi_{P_{k}}$ may not satisfy the BSPs, we are going to use the set $S_{P_{k}}$ as our chosen locations. The second step in our framework would then consist of constructing the appropriate distribution over assignments. Toward that end, consider the following LP.

$$
\begin{array}{lr}
\sum_{i \in S_{P_{k}}} x_{i, j}=1 & \forall j \in \mathcal{C} \\
z_{e, i} \geq x_{i, j}-x_{i, j^{\prime}} & \forall e=\left\{j, j^{\prime}\right\} \in \mathcal{P}, \forall i \in S_{P_{k}} \\
z_{e, i} \geq x_{i, j^{\prime}}-x_{i, j} & \forall e=\left\{j, j^{\prime}\right\} \in \mathcal{P}, \forall i \in S_{P_{k}} \\
z_{e}=\frac{1}{2} \sum_{i \in S_{P_{k}}} z_{e, i} & \forall e \in \mathcal{P} \\
z_{e} \leq \psi_{e} & \forall e \in \mathcal{P} \\
0 \leq x_{i, j}, z_{e}, z_{e, i} \leq 1 & \forall i \in S_{P_{k}}, \forall j \in \mathcal{C}, \forall e \in \mathcal{P} \tag{2.11}
\end{array}
$$

The variable $x_{i, j}$ can be interpreted as the probability of assigning point $j$ to location $i \in$ $S_{P_{k}}$. To understand the meaning of the $z$ variables, it is easier to think of the integral setting, where $x_{i, j}=1$ iff $j$ is assigned to $i$ and 0 otherwise. In this case, $z_{e, i}$ is 1 for $e=\left\{j, j^{\prime}\right\}$ iff exactly one of $j$ and $j^{\prime}$ are assigned to $i$. Thus, $z_{e}$ is 1 iff $j$ and $j^{\prime}$ are separated. We will later show that in the fractional setting $z_{e}$ is a lower bound on the probability that $j$ and $j^{\prime}$ are separated. Therefore, constraint (2.6) simply states that every point must be assigned to a center, and given the previous discussion, (2.10) expresses the provided BSPs.

Depending on which exact objective function we optimize, we must augment LP (2.6)(2.11) accordingly.

- $k$-center: Here we assume w.l.o.g. that the optimal radius $\tau_{B S P}^{*}$ of the original $k$-centerBSP instance is known. Observe that this value is always the distance between some point and some location, and hence there are only polynomially many alternatives for it. Thus, we execute our algorithm for each of those, and in the end keep the outcome that resulted in a feasible solution of minimum value. Given $\tau_{B S P}^{*}$, we add the following constraint to the LP (2.6)-(2.11).

$$
\begin{equation*}
x_{i, j}=0, \forall i, j: d(i, j)>\tau_{P_{k}}+2 \cdot \tau_{B S P}^{*} \tag{2.12}
\end{equation*}
$$

- $k$-median $(p=1) / k$-means $(p=2)$ : In this case, we augment the LP with the following objective function.

$$
\begin{equation*}
\min \sum_{j \in \mathcal{C}} \sum_{i \in S_{P_{k}}} x_{i, j} \cdot d(i, j)^{p} \tag{2.13}
\end{equation*}
$$

```
Algorithm 2: Approximating \(P_{k}\)-BSP
    \(\left(S_{P_{k}}, \phi_{P_{k}}\right) \leftarrow A_{P_{k}}(\mathcal{C}, \mathcal{F})\);
    Solve LP (2.6)-(2.11) with (2.12) for \(k\)-center, and with (2.13) for \(k\)-median \(/ k\)-means,
        and get a fractional solution \((\bar{x}, \bar{z})\);
    \(\phi \leftarrow \mathbf{K T}-\operatorname{Round}\left(\mathcal{C}, S_{P_{k}}, \mathcal{P}, \bar{x}, \bar{z}\right) ;\)
```

The second step of our framework begins by solving the appropriate extended LP for each variant of $P_{k}$, in order to acquire a fractional solution $(\bar{x}, \bar{z})$ to that LP. Finally, the distribution $\mathcal{D}$ over assignments $\mathcal{C} \mapsto S_{P_{k}}$ is constructed by running KT-Round $\left(\mathcal{C}, S_{P_{k}}, \mathcal{P}, \bar{x}, \bar{z}\right)$. Notice that this will yield an assignment $\phi \sim \mathcal{D}$, where $\mathcal{D}$ results from the internal randomness of KT-Round. Our overall approach for solving $P_{k}$ - BSP is presented in Algorithm 2.

Theorem 2.1.4. Let $\tau_{B S P}^{*}$ the optimal value of the given $P_{k}$-BSP instance. Then Algorithm 2 guarantees that $\left|S_{P_{k}}\right| \leq k, \operatorname{Pr}_{\phi \sim \mathcal{D}}\left[\phi(j) \neq \phi\left(j^{\prime}\right)\right] \leq 2 \psi_{q} \forall e_{q} \in \mathcal{P}$ and

1. $P_{k}$ is $k$-center: $\operatorname{Pr}_{\phi \sim \mathcal{D}}\left[d(\phi(j), j) \leq 2 \tau_{B S P}^{*}+\tau_{P_{k}}\right]=1$, for all $j \in \mathcal{C}$.
2. $P_{k}$ is $k$-median $(p=1) / k$-means $(p=2):\left(\sum_{j \in \mathcal{C}} \mathbb{E}_{\phi \sim \mathcal{D}}\left[d(\phi(j), j)^{p}\right]\right)^{1 / p} \leq 2 \tau_{B S P}^{*}+\tau_{P_{k}}$.

Proof. At first, since $S_{P_{k}}$ results from running $A_{P_{k}}$, it must be the case that $\left|S_{P_{k}}\right| \leq k$.
Focus now on LP (2.6)-(2.11) with either (2.12) or (2.13), depending on the underlying objective. In addition, let $S^{*}$ and $\mathcal{D}^{*}$ be the set of locations and the distribution over assignments $\mathcal{C} \mapsto S^{*}$, that constitute the optimal solution of $P_{k}$-BSP. Given those, let $x_{i, j}^{*}=\operatorname{Pr}_{\phi \sim \mathcal{D}^{*}}[\phi(j)=i]$ for all $i \in S^{*}$ and all $j \in \mathcal{C}$. Moreover, for every $i^{\prime} \in S^{*}$ let $\kappa\left(i^{\prime}\right)=\arg \min _{i \in S_{P_{k}}} d\left(i, i^{\prime}\right)$ by breaking ties arbitrarily. Finally, for all $i \in S_{P_{k}}$ we define $N(i)=\left\{i^{\prime} \in S^{*} \mid i=\kappa\left(i^{\prime}\right)\right\}$, and notice that the sets $N(i)$ form a partition of $S^{*}$.

Consider now the vectors $\hat{x}_{i, j}=\sum_{i^{\prime} \in N(i)} x_{i^{\prime}, j}^{*}$ for every $i \in S_{P_{k}}$ and $j \in \mathcal{C}$, and $\hat{z}_{e, i}=$ $\left|\hat{x}_{i, j}-\hat{x}_{i, j^{\prime}}\right|$ for every $e=\left\{j, j^{\prime}\right\} \in \mathcal{P}$ and $i \in S_{P_{k}}$. We first show that the above vectors constitute
a feasible solution of LP (2.6)-(2.11). Initially, notice that constraints (2.7), (2.8), (2.9), (2.11) are trivially satisfied. Regarding constraint (2.6), for any $j \in \mathcal{C}$ we have:

$$
\sum_{i \in S_{P_{k}}} \hat{x}_{i, j}=\sum_{i \in S_{P_{k}}} \sum_{i^{\prime} \in N(i)} x_{i^{\prime}, j}^{*}=\sum_{i^{\prime} \in S^{*}} x_{i^{\prime}, j}^{*}=1
$$

The second equality follows because the sets $N(i)$ induce a partition of $S^{*}$. The last equality is due to the optimal solution $\mathcal{D}^{*}, S^{*}$ satisfying $\sum_{i \in S^{*}} \operatorname{Pr}_{\phi \sim \mathcal{D}^{*}}[\phi(j)=i]=1$.

To show satisfaction of constraint (2.10) focus on any $e=\left\{j, j^{\prime}\right\} \in P^{\prime}$ and $i \in S_{P_{k}}$. At first, it is clear that:

$$
\hat{z}_{e, i}=\left|\sum_{i^{\prime} \in N(i)}\left(x_{i^{\prime}, j}^{*}-x_{i^{\prime}, j^{\prime}}^{*}\right)\right| \leq \sum_{i^{\prime} \in N(i)}\left|x_{i^{\prime}, j}^{*}-x_{i^{\prime}, j^{\prime}}^{*}\right|
$$

Therefore, we can easily upper bound $\hat{z}_{e}$ as follows:

$$
\begin{align*}
\hat{z}_{e} & =\frac{1}{2} \sum_{i \in S_{P_{k}}} \hat{z}_{e, i} \\
& \leq \frac{1}{2} \sum_{i \in S_{P_{k}}} \sum_{i^{\prime} \in N(i)}\left|x_{i^{\prime}, j}^{*}-x_{i^{\prime}, j}^{*}\right| \\
& \leq \frac{1}{2} \sum_{i^{\prime} \in S^{*}}\left|x_{i^{\prime}, j}^{*}-x_{i^{\prime}, j}^{*}\right| \tag{2.14}
\end{align*}
$$

To move one, notice that:

$$
\begin{align*}
\operatorname{Pr}_{\phi \sim \mathcal{D}^{*}}\left[\phi(j)=\phi\left(j^{\prime}\right)\right] & =\sum_{i^{\prime} \in S^{*}} \operatorname{Pr}_{\phi \sim \mathcal{D}^{*}}\left[\phi(j)=i^{\prime} \wedge \phi\left(j^{\prime}\right)=i^{\prime}\right] \\
& \leq \sum_{i^{\prime} \in S^{*}} \min \left\{x_{i^{\prime}, j}^{*}, x_{i^{\prime}, j^{\prime}}^{*}\right\} \tag{2.15}
\end{align*}
$$

To relate (2.14) and (2.15) consider the following trick.

$$
\begin{align*}
& \sum_{i^{\prime} \in S^{*}} \min \left\{x_{i^{\prime}, j}^{*}, x_{i^{\prime}, j^{\prime}}^{*}\right\}+\frac{1}{2} \sum_{i^{\prime} \in S^{*}}\left|x_{i^{\prime}, j}^{*}-x_{i^{\prime}, j^{\prime}}^{*}\right|= \\
& \sum_{i^{\prime} \in S^{*}}\left(\min \left\{x_{i^{\prime}, j}^{*}, x_{i^{\prime}, j^{\prime}}^{*}\right\}+\frac{\left|x_{i^{\prime}, j}^{*}-x_{i^{\prime}, j^{\prime}}^{*}\right|}{2}\right)= \\
& \sum_{i^{\prime} \in S^{*}} \frac{x_{i^{\prime}, j}^{*}+x_{i^{\prime}, j^{\prime}}^{*}}{2}=\frac{2}{2}=1 \tag{2.16}
\end{align*}
$$

Finally, combining (2.14),(2.15),(2.16) we get:

$$
\begin{aligned}
\hat{z}_{e} & \leq 1-\sum_{i^{\prime} \in S^{*}} \min \left\{x_{i^{\prime}, j}^{*}, x_{i^{\prime}, j^{\prime}}^{*}\right\} \\
& \leq \operatorname{Pr}_{\phi \sim \mathcal{D}^{*}}\left[\phi(j) \neq \phi\left(j^{\prime}\right)\right] \leq \psi_{e}
\end{aligned}
$$

where the last inequality follows from optimality of $\mathcal{D}^{*}$.
Now that we know that $(\hat{x}, \hat{z})$ is a feasible solution for (2.6)-(2.11), we proceed by considering how this solution affects the objective function of each underlying problem.
$k$-center: The objective here is captured by the additional constraint (2.12). Hence, we also need to show that $\hat{x}$ satisfies (2.12), i.e., that for all $i \in S_{P_{k}}, j \in \mathcal{C}$ for which $d(i, j)>$ $\tau_{P_{k}}+2 \cdot \tau_{B S P}^{*}$, we have $\hat{x}_{i, j}=0$.

Suppose for the sake of contradiction that there exists a $j \in \mathcal{C}$ and an $i \in S_{P_{k}}$ such that $d(i, j)>\tau_{P_{k}}+2 \cdot \tau_{B S P}^{*}$ and $\hat{x}_{i, j}>0$. Since $\hat{x}_{i, j}=\sum_{i^{\prime} \in N(i)} x_{i^{\prime}, j}^{*}$, this implies that there exists $i^{\prime} \in N(i)$ with $x_{i^{\prime}, j}^{*}>0$, which consecutively implies $d\left(i^{\prime}, j\right) \leq \tau_{B S P}^{*}$. By the triangle inequality:

$$
\begin{equation*}
d(i, j) \leq d\left(i, i^{\prime}\right)+d\left(i^{\prime}, j\right) \leq d\left(i, i^{\prime}\right)+\tau_{B S P}^{*} \tag{2.17}
\end{equation*}
$$

Moving forward, we get:

$$
\begin{equation*}
d\left(i, i^{\prime}\right) \leq d\left(i^{\prime}, \phi_{P_{k}}(j)\right) \leq d\left(i^{\prime}, j\right)+d\left(\phi_{P_{k}}(j), j\right) \leq \tau_{B S P}^{*}+\tau_{P_{k}} \tag{2.18}
\end{equation*}
$$

Combining (2.17),(2.18) gives the desired contradiction.
$k$-median $(p=1) / k$-means $(p=2)$ : Here the objective function for $\hat{x}$ is given by:

$$
\begin{equation*}
\left(\sum_{j \in \mathcal{C}} \sum_{i \in S_{P_{k}}} \hat{x}_{i, j} d(i, j)^{p}\right)^{\frac{1}{p}}=\left(\sum_{j \in \mathcal{C}} \sum_{i \in S_{P_{k}}} \sum_{i^{\prime} \in N(i)} x_{i^{\prime}, j}^{*} d(i, j)^{p}\right)^{\frac{1}{p}} \tag{2.19}
\end{equation*}
$$

In addition, for $i^{\prime} \in N(i)$ we also get:

$$
\begin{align*}
d(i, j) & \leq d\left(i, i^{\prime}\right)+d\left(i^{\prime}, j\right) \leq d\left(i^{\prime}, \phi_{P_{k}}(j)\right)+d\left(i^{\prime}, j\right) \\
& \leq d\left(i^{\prime}, j\right)+d\left(\phi_{P_{k}}(j), j\right)+d\left(i^{\prime}, j\right) \leq 2 d\left(i^{\prime}, j\right)+d\left(\phi_{P_{k}}(j), j\right) \tag{2.20}
\end{align*}
$$

Combining (2.19), (2.20) and the fact that the median and means objectives are monotone norms, we get $(2.19) \leq A+B$, where:

$$
\begin{align*}
A & =\left(\sum_{j \in \mathcal{C}} \sum_{i \in S_{P_{k}}} \sum_{i^{\prime} \in N(i)} 2 x_{i^{\prime}, j}^{*} d\left(i^{\prime}, j\right)^{p}\right)^{\frac{1}{p}}=2\left(\sum_{j \in \mathcal{C}} \sum_{i^{\prime} \in S^{*}} x_{i^{\prime}, j}^{*} d\left(i^{\prime}, j\right)^{p}\right)^{\frac{1}{p}} \leq 2 \tau_{B S P}^{*}  \tag{2.21}\\
B & =\left(\sum_{j \in \mathcal{C}} \sum_{i \in S_{P_{k}}} \sum_{i^{\prime} \in N(i)} x_{i^{\prime}, j}^{*} d\left(\phi_{P_{k}}(j), j\right)^{p}\right)^{\frac{1}{p}}=\left(\sum_{j \in \mathcal{C}} d\left(\phi_{P_{k}}(j), j\right)^{p} \sum_{i \in S_{P_{k}}} \sum_{i^{\prime} \in N(i)} x_{i^{\prime}, j}^{*}\right)^{\frac{1}{p}} \\
& =\left(\sum_{j \in \mathcal{C}} d\left(\phi_{P_{k}}(j), j\right)^{p} \sum_{i^{\prime} \in S^{*}} x_{i^{\prime}, j}^{*}\right)^{\frac{1}{p}}=\left(\sum_{j \in \mathcal{C}} d\left(\phi_{P_{k}}(j), j\right)^{p}\right)^{\frac{1}{p}}=\tau_{P_{k}} \tag{2.22}
\end{align*}
$$

Combining (2.21), (2.22) we finally get $(2.19) \leq 2 \tau_{B S P}^{*}+\tau_{P_{k}}$.
Since $(\hat{x}, \hat{z})$ is a feasible solution to the appropriate version of the assignment LP, step 2 of

Algorithm 2 is well-defined, and thus can compute a solution $(\bar{x}, \bar{z})$ that satisfies (2.6)-(2.11) and additionally: i)

$$
\begin{equation*}
\bar{x}_{i, j}=0, \forall i, j: d(i, j)>\tau_{P_{k}}+2 \cdot \tau_{B S P}^{*} \tag{2.23}
\end{equation*}
$$

for $k$-center, and ii)

$$
\begin{align*}
\left(\sum_{j \in \mathcal{C}} \sum_{i \in S_{P_{k}}} \bar{x}_{i, j} d(i, j)^{p}\right)^{\frac{1}{p}} & \leq\left(\sum_{j \in \mathcal{C}} \sum_{i \in S_{P_{k}}} \hat{x}_{i, j} d(i, j)^{p}\right)^{\frac{1}{p}}  \tag{2.24}\\
& \leq 2 \tau_{B S P}^{*}+\tau_{P_{k}} \tag{2.25}
\end{align*}
$$

for $k$-median $(p=1) / k$-means $(p=2)$.
Because $(\bar{x}, \bar{z})$ satisfies (2.6), (2.7), (2.8), (2.9), (2.11), KT-Round can be applied for $V=$ $\mathcal{C}, L=S_{P_{k}}, E=\mathcal{P}$. Let $\phi$ be the assignment returned by KT-Round $\left(\mathcal{C}, S_{P_{k}}, \mathcal{P}, \bar{x}, \bar{z}\right)$, and $\mathcal{D}$ the distribution representing the internal randomness of this process. From Theorem 2.1.3 we have $\operatorname{Pr}_{\phi \sim \mathcal{D}}\left[\phi(j) \neq \phi\left(j^{\prime}\right)\right] \leq 2 \bar{z}_{e}, \forall e=\left\{j, j^{\prime}\right\} \in \mathcal{P}$. Hence, for every $e \in \mathcal{P}$ we have

$$
\operatorname{Pr}_{\phi \sim \mathcal{D}}\left[\phi(j) \neq \phi\left(j^{\prime}\right)\right] \leq 2 \bar{z}_{e} \leq 2 \psi_{e}
$$

because $\bar{z}$ satisfies (2.10).

Regarding all the different objective functions, we have the following. For $k$-center, because of (2.23) and the second property of Theorem 2.1.3, we know that a point $j \in \mathcal{C}$ will never be assigned to a location $i \in S_{P_{k}}$, such that $d(i, j)>\tau_{P_{k}}+2 \tau_{B S P}^{*}$. Therefore, $\operatorname{Pr}_{\phi \sim \mathcal{D}}[d(\phi(j), j) \leq$
$\left.\tau_{P_{k}}+2 \tau_{B S P}^{*}\right]=1$ for all $j \in \mathcal{C}$. As for the $k$-median $(p=1) / k$-means $(p=2)$ objectives, the second property of Theorem 2.1.3 ensures:

$$
\begin{aligned}
\left(\sum_{j \in \mathcal{C}} \mathbb{E}_{\phi \sim \mathcal{D}}\left[d(\phi(j), j)^{p}\right]\right)^{1 / p} & =\left(\sum_{j \in \mathcal{C}} \sum_{i \in S_{P_{k}}} \operatorname{PrD}_{\phi \sim \mathcal{D}}[\phi(j)=i] \cdot d(i, j)^{p}\right)^{1 / p} \\
& =\left(\sum_{j \in \mathcal{C}} \sum_{i \in S_{P_{k}}} \bar{x}_{i, j} \cdot d(i, j)^{p}\right)^{1 / p} \leq 2 \tau_{B S P}^{*}+\tau_{P_{k}}
\end{aligned}
$$

where the last inequality follows from (2.25).

Since $P_{k}$ is a less restricted version of $P_{k}$ - BSP , the optimal solution value $\tau_{P_{k}}^{*}$ for $P_{k}$ in the original instance where we dropped the BSPs, should satisfy $\tau_{P_{k}}^{*} \leq \tau_{B S P}^{*}$. Therefore, because $A_{P_{k}}$ is a $\rho$-approximation algorithm for $P_{k}$, we get $\tau_{P_{k}} \leq \rho \cdot \tau_{B S P}^{*}$. The latter implies the following.

Corollary 2.1.5. The approximation ratio achieved through Algorithm 2 is $(\rho+2)$ for all $k$ -center-BSP, $k$-median-BSP and $k$-means-BPS.

### 2.1.2.3 Better Algorithm for $k$-center with Must-Link Constraints

Since must-link constraints (ML) are a special case of BSPs, Algorithm 2 provides approximation results for the former as well (also note that due to $\psi_{p}=0 \forall p$, we have no pairwise constraint violation when using Algorithm 2 purely for ML). However, in this section we demonstrate how we can get improved approximation guarantees for $k$-center with ML constraints. Specifically, we provide a 3-approximation for this problem, which constitutes a clear improvement over the 5 -approximation, that comes when Algorithm 2 is executed using the best approximation algorithm for $k$-center [8, 98] (approximation ratio of 2).

To begin with the description of our algorithm, recall that in the ML case we are only

```
Algorithm 3: Approximating ML Constraints
    \(C \leftarrow \emptyset ;\)
    \(S \leftarrow \emptyset ;\)
    Initially all \(C_{1}, C_{2}, \ldots, C_{t}\) are considered uncovered;
    while there exists an uncovered \(C_{q} \mathbf{d o}\)
        Pick an uncovered \(C_{q}\);
        Pick an arbitrary point \(j_{q} \in C_{q}\);
        \(C \leftarrow C \cup\left\{j_{q}\right\} ;\)
        \(C_{q}\) and all neighboring cliques \(C_{p}\) of it are now considered covered;
    end
    for all \(j_{q} \in C\) do
        \(i_{q} \leftarrow \arg \min _{i \in \mathcal{F}} d\left(i, j_{q}\right) ;\)
        \(S \leftarrow S \cup\left\{i_{q}\right\} ;\)
    end
    for all \(j \in \mathcal{C}\) do
        Let \(j_{q} \in C\) the point whose clique \(C_{q}\) covered \(j\) 's clique in the first while loop;
        \(\phi(j) \leftarrow i_{q} ;\)
    end
```

looking for a set of locations $S$ and an assignment $\phi: \mathcal{C} \mapsto S$, and not for a distribution over assignments. Also, notice that the must-link relation is transitive. If for $j, j^{\prime}$ we want $\phi(j)=$ $\phi\left(j^{\prime}\right)$, and for $j^{\prime}, j^{\prime \prime}$ we also require $\phi\left(j^{\prime}\right)=\phi\left(j^{\prime \prime}\right)$, then $\phi(j)=\phi\left(j^{\prime \prime}\right)$ is necessary as well. Given that, we view the input as a partition $C_{1}, C_{2}, \ldots, C_{t}$ of the points of $\mathcal{C}$, where all points in $C_{q}$, with $q \in\{1, \ldots, t\}$, must be assigned to the same location of $S$. We call each part $C_{i}$ of this partition a clique. Finally, we can once more assume w.l.o.g. that the optimal radius $\tau^{*}$ is known. Before we proceed with Algorithm 3 that demonstrates all necessary details of our approach, we need the following definition.

Definition 2.1.6. Two cliques $C_{q}, C_{p}$ are neighboring if $d\left(j, j^{\prime}\right) \leq 2 \tau^{*}, \forall j \in C_{q}, \forall j^{\prime} \in C_{p}$.

Theorem 2.1.7. Algorithm 3 is a 3-approximation for $k$-center with ML constraints.

Proof. Initially, observe that the must-link constraints are satisfied. When the algorithm chooses
a location $i_{q}$ based on some $j_{q} \in C$, all the points in $C_{q}$ are assigned to $i_{q}$. Also, for the neighboring cliques of $C_{q}$ that got covered by it, their whole set of points ends up assigned to $i_{q}$.

We now argue about the achieved approximation ratio. For one thing, it is clear that for every $j, j^{\prime} \in C_{q}$ we must have $d\left(j, j^{\prime}\right) \leq 2 \tau^{*}$, and therefore $C_{q}$ is a neighboring clique of itself. For each $j_{q} \in C$ we choose a location $i_{q}$ such that $d\left(i_{q}, j_{q}\right) \leq \tau^{*}$. Also, all points $j$ assigned to $i_{q}$ belong to neighboring cliques of $C_{q}$, and therefore $d\left(i_{q}, j\right) \leq d\left(i_{q}, j_{q}\right)+d\left(j_{q}, j\right) \leq 3 \tau^{*}$.

Finally, we need to show that either the cardinality or the knapsack constraint on the set of chosen locations is satisfied. Toward that end, notice that if $C_{q}$ and $C_{p}$ belong in the same cluster in the optimal solution, then they are neighboring. Say $i^{\star}$ is the location they are both assigned to. Then for all $j \in C_{q}$ and $j^{\prime} \in C_{p}$ we get $d\left(j, i^{\star}\right) \leq \tau^{*}$ and $d\left(j^{\prime}, i^{\star}\right) \leq \tau^{*}$. Hence, by the triangle inequality for all $j \in C_{q}$ and $j^{\prime} \in C_{p}$ we have $d\left(j, j^{\prime}\right) \leq 2 \tau^{*}$.

Given the previous observation, it must be the case that for every $j_{q} \in C$, the optimal solution assigns it to a location $i_{j_{q}}^{*}$, such that for every other $j_{q^{\prime}} \in C$ we have $i_{j_{q}}^{*} \neq i_{j_{q^{\prime}}}^{*}$. Therefore, in the presence of a cardinality constraint $|S| \leq|C|=\sum_{i_{j q}^{*}: j_{q} \in C} 1 \leq k$.

### 2.1.3 Experimental Results for PAIRFAIRCLU

Here we present some experimental results, which first appeared in [38], and involve implementations of both our algorithms for PairFairclu.

We implement our algorithms in Python 3.8 and run our experiments on AMD Opteron 6272 @ 2.1 GHz with 64 cores and 512 GB 1333 MHz DDR3 memory. The first goal of the experiments at hand is to compare our approach from [38] with the $k$-means algorithm of Anderson et al. [50]; recall that this paper considers a very similar individually-fair model
(see Section 1.5.1.2). The second goal of these experiments is to compare our $k$-center algorithm from [38] with our earlier work from [37]. Our code is publicly available at https: //github.com/chakrabarti/pairwise_constrained_clustering.

Datasets: We use 3 datasets from the UCI ML Repository [114]: (1) Bank-4,521 points [115], (2) Adult-32,561 points [116], and (3) Creditcard-30,000 points [117].

Algorithms: In all of our experiments $\mathcal{C}=\mathcal{F}$. When solving PairFairClu with the $k$ means objective, we use Lloyd's algorithm [118] in the first step of Algorithm 2, and in return we get a set of points $L$. The set of chosen locations $S$ is then constructed by getting the nearest point in $\mathcal{C}$ for every point of $L$. This is exactly the approach used by Anderson et al. [50], whose overall algorithm we denote by ALG-IF. To compare Algorithm 2 with ALG-IF we need a deterministic solution in the output. To achieve the latter for ALG-IF we use independent sampling; this fixes the assignment of each $j \in \mathcal{C}$ to some $i \in S$, based on the distribution $\phi_{j}$ produced by ALG-IF. For PAIRFAIRCLU with the $k$-center objective, we use binary search to compute a lower bound $\tau_{C}^{*}$ for the value of the optimal solution. We will be referring to the LP-based algorithm from [38] as ALG-LP, and to the algorithm from [37] as ALG-F.

Fairness Constraints: We consider three similarity metrics ( $F_{1}, F_{2}, F_{3}$ ) for generating the $\psi$ values. We use $F_{1}$ when addressing the $k$-center objective and $F_{2}, F_{3}$ for $k$-means. $F_{1}$ is the metric used in our simulations from [37] and $F_{2}, F_{3}$ are the metrics used in the experimental evaluation of the algorithms of Anderson et al. [50].
$F_{1}$ sets $\psi_{j, j^{\prime}}$ to $16 \cdot d\left(j, j^{\prime}\right) / R_{S c r}$ if $16 \cdot d\left(j, j^{\prime}\right) \leq R_{S c r}$, where $R_{S c r}$ is the radius given by running the Scr algorithm [119] on the provided input.
$F_{2}$ is defined in [50] so that the separation probability between a pair $j, j^{\prime}$ is given by $d\left(j, j^{\prime}\right)$, scaled linearly to ensure all such probabilities are in $[0,1]$. Adopting the exact approach taken by

|  | k | 4 | 6 | 8 | 10 |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Adult | Alg-LP | 3.65 | 7.95 | 17.57 | 28.83 |
|  | ALG-IF | 91.60 | 99.81 | 96.16 | 99.81 |
| Bank | Alg-LP | 0.29 | 1.13 | 0.86 | 1.49 |
|  | ALG-IF | 54.87 | 60.52 | 97.04 | 85.36 |
| Credit | Alg-LP | 1.85 | 15.28 | 20.53 | 31.08 |
|  | ALG-IF | 68.00 | 99.41 | 99.7 | 100.00 |

Table 2.1: Percentage of constraints that are violated on average for metric $F_{2}$

Anderson et al. [50] when using this metric, we only consider pairwise constraints between each $j$ and its closest $m$ neighbors. For our experiments, we set $m=100$.

Another similarity metric used by Anderson et al. [50] is the one we call $F_{3}$. For any $j \in \mathcal{C}$, let $r_{j}$ the minimum distance such that $\left|j^{\prime} \in \mathcal{C}: d\left(j, j^{\prime}\right) \leq r_{j}\right| \geq|\mathcal{C}| / k$. Then, according to $F_{3}$, the similarity value between some $j$ and any other $j^{\prime}$ with $d\left(j, j^{\prime}\right) \leq r_{j}$, is set to $d\left(j, j^{\prime}\right) / r_{j}$.

Implementation Details: As performed in [37, 50], we uniformly sample $N$ points from each dataset and run all algorithms on those sets, while only considering a subset of the numerical attributes and normalizing the features to have zero mean and unit variance. In our comparisons with [50] we use $N=500$, while in the comparisons with [37] $N$ is set to 250 . For the number of clusters $k$, we study the values $\{4,6,8,10\}$ when comparing to [50], and $\{30,40,50,60\}$ when comparing to [37]. Finally, to estimate the empirical separation probabilities and the underlying objective function cost, we run 5000 trials for each randomized assignment procedure, and then compute averages for the performance measures we are interested in.

Comparison with Anderson et al. [50]: In Tables 2.1 and 2.2, we show what percentage of fairness constraints are violated by ALG-IF and our algorithm, for the fairness constraints induced by $F_{2}$ and $F_{3}$, allowing for an $\epsilon=0.05$ threshold on the violation of a separation probability

|  | k | 4 | 6 | 8 | 10 |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Adult | Alg-LP | 0.09 | 0.28 | 0.61 | 0.97 |
|  | ALG-IF | 8.16 | 6.09 | 7.54 | 9.80 |
| Bank | Alg-LP | 0.02 | 0.18 | 0.35 | 0.56 |
|  | ALG-IF | 3.84 | 4.23 | 7.02 | 6.45 |
| Credit | Alg-LP | 0.00 | 0.25 | 0.24 | 0.37 |
|  | ALG-IF | 1.01 | 4.05 | 4.66 | 4.07 |

Table 2.2: Percentage of constraints that are violated on average for metric $F_{3}$
bound; we only consider a pair's fairness constraint to be violated if the empirical probability of them being separated exceeds that set by the fairness metric by more than $\epsilon$. It is clear that our algorithm outperforms ALG-IF consistently across different values of $k$, different datasets, and both types of similarity defining metrics.

In order to compare the objective value achieved by both algorithms, we first compute the average connection costs over the 5000 runs. Since the cost of the clustering returned by Lloyd's algorithm contributes to both ALG-LP and ALG-IF, we utilize that as an approximation of the cost of fairness. In other words, we divide the objective value of the final solutions by the cost of the clustering produced by Lloyd, and call this quantity cost of fairness. The corresponding comparisons are presented in Tables 2.3, 2.4. The cost of fairness for both algorithms is very similar, demonstrating a clear advantage of ALG-LP, since it dominates ALG-IF in the percentage of fairness constraints violated.

Comparison with Brubach et al. [37]: Before we proceed with our analysis, we state that the reason for choosing $F_{1}$ for this set of comparisons, is that this particular metric resulted in the best possible objective function values in our simulations from [37]. In Table 2.5 we show what percentage of fairness constraints are violated by oALG-F and ALG-LP, using an $\epsilon=0$; if

|  | k | 4 | 6 | 8 | 10 |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Adult | Alg-LP | 1.81 | 2.36 | 3.12 | 3.53 |
|  | ALG-IF | 1.92 | 2.35 | 3.13 | 3.44 |
| Bank | Alg-LP | 2.30 | 3.34 | 4.07 | 5.13 |
|  | ALG-IF | 2.39 | 3.63 | 4.55 | 5.10 |
| Credit | Alg-LP | 1.86 | 2.22 | 2.53 | 2.53 |
|  | ALG-IF | 1.84 | 2.18 | 2.49 | 2.50 |

Table 2.3: Cost of fairness for metric $F_{2}$

|  | k | 4 | 6 | 8 | 10 |
| :---: | :--- | :--- | :--- | :--- | :--- |
| Adult | Alg-LP | 1.12 | 1.18 | 1.23 | 1.22 |
|  | ALG-IF | 1.13 | 1.18 | 1.24 | 1.23 |
| Bank | Alg-LP | 1.25 | 1.39 | 1.46 | 1.55 |
|  | ALG-IF | 1.32 | 1.63 | 1.70 | 1.91 |
| Credit | Alg-LP | 1.13 | 1.11 | 1.14 | 1.06 |
|  | ALG-IF | 1.11 | 1.11 | 1.14 | 1.06 |

Table 2.4: Cost of fairness for metric $F_{3}$
the empirical probability of separation of a pair exceeds the bound set by the fairness metric by any amount, it is considered a violation. Both ALG-F and ALG-LP mostly lead to 0 violations, with our algorithm producing a small number of violations in a few cases, which are essentially negligible. In Table 2.6 we show the cost of the clusterings produced by ALG-F and ALG-LP, measured in the normalized metric space by taking the average of the maximum radius of any cluster over the 5000 runs. Our algorithm in all cases produces clusterings with significantly smaller radii.

Runtime: The average runtime of ALG-LP over all datasets was $\mathrm{k}=30: 140 \mathrm{~s} / \mathrm{k}=40$ : 150s $/ \mathrm{k}=50: 160 \mathrm{~s} / \mathrm{k}=60: 160 \mathrm{~s}$ for $F_{1}, \mathrm{k}=4: 63 \mathrm{~s} / \mathrm{k}=6: 20000 \mathrm{~s} / \mathrm{k}=8: 32300 \mathrm{~s} / \mathrm{k}=10:$ 41000s for $F_{2}$, and $\mathrm{k}=4: 83 \mathrm{~s} / \mathrm{k}=6: 8400 \mathrm{~s} / \mathrm{k}=8: 8900 \mathrm{~s} / \mathrm{k}=10: 7600 \mathrm{~s}$ for $F_{3}$.

|  | k | 30 | 40 | 50 | 60 |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Adult | Alg-2 | 0.03 | 0.00 | 0.00 | 0.00 |
|  | ALG-F | 0.00 | 0.00 | 0.00 | 0.00 |
| Bank | Alg-2 | 0.00 | 0.04 | 0.06 | 0.07 |
|  | ALG-F | 0.00 | 0.00 | 0.00 | 0.00 |
| Credit | Alg-2 | 0.00 | 0.00 | 0.00 | 0.00 |
|  | ALG-F | 0.00 | 0.00 | 0.00 | 0.00 |

Table 2.5: Percentage of constraints that are violated on average for metric $F_{1}$

|  | k | 30 | 40 | 50 | 60 |
| :---: | :--- | :--- | :--- | :--- | :--- |
| Adult | Alg-2 | 0.23 | 0.20 | 0.17 | 0.16 |
|  | ALG-F | 0.46 | 0.46 | 0.43 | 0.42 |
| Bank | Alg-2 | 0.08 | 0.07 | 0.06 | 0.05 |
|  | ALG-F | 0.17 | 0.15 | 0.14 | 0.13 |
| Credit | Alg-2 | 0.25 | 0.24 | 0.21 | 0.19 |
|  | ALG-F | 0.43 | 0.43 | 0.41 | 0.41 |

Table 2.6: Objective achieved for metric $F_{1}$

### 2.2 Solving EQCenter

### 2.2.1 Structural Properties of the Problem

The purpose of this section is to answer questions regarding the combinatorial nature of our newly proposed fairness constraints, as studied in EQCENTER. As mentioned in the introduction, all our results here are for $k \geq 2$, since $k=1$ is a trivial case.

At first, we want to investigate the range of $\alpha$ for which our problems always admit a feasible solution. Ideally, an $\alpha$ value close to 1 would be the most fair, since such a value provides the most equitable outcomes. However, as the next theorem suggests, such a guarantee is impossible.


Figure 2.1: Construction for $m=8$. The solid lines represent a distance of 1 between the corresponding points. The dashed lines correspond to similarity sets. For instance, the dashed line between $j_{1}$ and $j_{9}$ shows that $\pi\left(j_{1}\right)=j_{9}$ and $\pi\left(j_{9}\right)=j_{1}$.

Theorem 2.2.1. For both EQCENTER-PP and EQCENTER-AG, there exist instances with $\alpha<2$ that do not admit any feasible solution.

Proof. Let $m$ be a very large even integer, with $\frac{m}{2}$ also being an even integer. We consider $2 m$ points $\mathcal{C}=\left\{j_{1}, j_{2}, \ldots, j_{2 m-1}, j_{2 m}\right\}$ in a cycle, where $d\left(j_{i}, j_{i+1}\right)=1$ for all $i \in[2 m-1]$, and also $d\left(j_{2 m}, j_{1}\right)=1$. The rest of the distances are set to be the shortest path ones, based on those already defined. This is a valid metric space, since it constitutes the shortest path metric resulting from a simple cycle graph of $2 m$ vertices.

To construct the similarity sets, we map each point $j$ to another point $\pi(j) \neq j$, such that the function $\pi: \mathcal{C} \mapsto \mathcal{C}$ is one-to-one and $\pi(\pi(j))=j$. Given that, the similarity set of point $j$ will be set to be $\mathcal{S}_{j}=\{\pi(j)\}$. Now let $\mathcal{C}_{1}=\left\{j_{i} \mid i\right.$ is odd $\}$ and $\mathcal{C}_{2}=\left\{j_{i} \mid i\right.$ is even $\}$. For
every odd $i \in[m]$, set $\pi\left(j_{i}\right)=j_{i+m}$ and $\pi\left(j_{i+m}\right)=j_{i}$. In this way, because $m$ is even, we map every point of $\mathcal{C}_{1}$ to some other point of $\mathcal{C}_{1}$. Also, note that for every $j \in \mathcal{C}_{1}$ we will have $d(j, \pi(j))=m$. For the points $j_{i} \in \mathcal{C}_{2}$, consider them in increasing order of $i$. If $j_{i}$ is not already mapped to some other point, set $\pi\left(j_{i}\right)=j_{i+\frac{m}{2}}$ and $\pi\left(j_{i+\frac{m}{2}}\right)=j_{i}$. This is a valid assignment because $\frac{m}{2}$ is assumed to be an even integer. At the end of the above process, we have created a one-to-one mapping between the points of $\mathcal{C}_{2}$, such that for every $j \in \mathcal{C}_{2}$ we have $d(j, \pi(j))=\frac{m}{2}$. This concludes the description of the similarity sets. Finally, this pairing process for $\mathcal{C}_{1}$ and $\mathcal{C}_{2}$ is possible, because both sets include an even number of points. See Figure 2.1 for an example.

To conclude the description of the input we also assume that $k=2$. In addition, note that because for all $j$ we have $\left|\mathcal{S}_{j}\right|=1$, constraints (1.1) and (1.2) are equivalent and hence showing infeasibility for this instance covers both EQCEnter-PP and EQCEnter-AG. Finally, to prove the statement of the theorem, it suffices to show that for all possible choices of centers and all possible corresponding assignments $\phi$, there will always be a point $j_{p}$ for which $d\left(j_{p}, \phi\left(j_{p}\right)\right) \geq$ $2 d\left(\pi\left(j_{p}\right), \phi\left(\pi\left(j_{p}\right)\right)\right)$.

At first, notice that there exists no feasible solution that uses just one center. Supposing otherwise, let $c$ be the only chosen center. Then there exists only one possible assignment for $c$, and that is $\phi(c)=c$. Hence $d(c, \phi(c))=0$, and the constraint for $\pi(c)$ will never be satisfied.

Now we will show that even solutions that pick two centers $c_{1}, c_{2}$ cannot admit any feasible assignment. We proceed via a case analysis on $d\left(c_{1}, c_{2}\right)$.

- $d\left(c_{1}, c_{2}\right) \leq \frac{m}{3}$ : Because the points of $\mathcal{C}_{1}$ and $\mathcal{C}_{2}$ alternate in the metric cycle, we know that there exists a $j \in \mathcal{C}_{1}$ such that $d\left(j, c_{1}\right) \leq 1$ (in the example of Figure 2.1 we might have $c_{1}=j_{2}, c_{2}=j_{3}$ and $j=j_{1}, \pi(j)=j_{9}$ ). By the triangle inequality we also get
$d\left(j, c_{2}\right) \leq \frac{m}{3}+1$. As for the point $\pi(j)$, we have:

$$
\begin{aligned}
& d\left(\pi(j), c_{1}\right) \geq d(\pi(j), j)-d\left(j, c_{1}\right) \geq m-1 \\
& d\left(\pi(j), c_{2}\right) \geq d(\pi(j), j)-d\left(j, c_{2}\right) \geq m-\frac{m}{3}-1=\frac{2 m}{3}-1
\end{aligned}
$$

From $\pi(j)$ 's perspective, the best case situation regarding its fairness constraint is if $\pi(j)$ gets assigned to its closest center, and $j$ gets assigned to its farthest one. Given all the previous inequalities, we see that the best possible service for $\pi(j)$ is $\frac{2 m}{3}-1$, and the worst possible service for $j$ is $\frac{m}{3}+1$. We next show that even in this ideal situation for $\pi(j)$, its fairness constraint with $\alpha<2$ will never be satisfied if $m$ is significantly large. To see this, note that $\frac{2 m / 3-1}{m / 3+1}$ is an increasing function of $m$ and also:

$$
\lim _{m \rightarrow \infty}\left(\frac{\frac{2 m}{3}-1}{\frac{m}{3}+1}\right)=\frac{2 / 3}{1 / 3}=2
$$

Therefore, for every given $\alpha<2$, there exists an $m_{a}$ such that $\frac{2 m_{a} / 3-1}{m_{a} / 3+1}>\alpha$.

- $\frac{m}{3}<d\left(c_{1}, c_{2}\right) \leq \frac{2 m}{3}$ : In this case, because $m$ is assumed to be significantly large and because the points of $\mathcal{C}_{1}, \mathcal{C}_{2}$ alternate in the metric cycle, we can find a point $j \in \mathcal{C}_{1}$ in the shortest path between $c_{1}$ and $c_{2}$, which will be approximately in the middle of the path. Letting $\gamma \in\left(\frac{1}{3}, \frac{2}{3}\right]$ such that $d\left(c_{1}, c_{2}\right)=\gamma m$, we have

$$
\frac{\gamma m}{2}-1 \leq d\left(j, c_{1}\right), d\left(j, c_{2}\right) \leq \frac{\gamma m}{2}+1
$$

(in the example of Figure 2.1 we might have $c_{1}=j_{1}, c_{2}=j_{5}$ and $j=j_{3}, \pi(j)=j_{11}$ ).

Regarding the possible assignments for $\pi(j)$ we have:

$$
\begin{aligned}
& d\left(\pi(j), c_{1}\right) \geq d(j, \pi(j))-d\left(j, c_{1}\right) \geq m-\frac{\gamma m}{2}-1=m\left(\frac{2-\gamma}{2}\right)-1 \\
& d\left(\pi(j), c_{2}\right) \geq d(j, \pi(j))-d\left(j, c_{2}\right) \geq m-\frac{\gamma m}{2}-1=m\left(\frac{2-\gamma}{2}\right)-1
\end{aligned}
$$

Again we will focus on the best case situation for $\pi(j)$, which according to the previous analysis is $\pi(j)$ getting assigned to a center at distance $\frac{m(2-\gamma)}{2}-1$ from it, and $j$ getting assigned to a center at distance $\frac{\gamma m}{2}+1$. Therefore, we consider the ratio $\frac{m(2-\gamma) / 2-1}{\gamma m / 2+1}$, and we are going to prove that even in this ideal case for $\pi(j)$, its fairness constraint for $\alpha<2$ will not be satisfiable if $m$ is sufficiently large. At first, because $\frac{2-\gamma}{2}, \frac{\gamma}{2}>0$ the previous ratio will be an increasing function of $m$. In addition,

$$
\lim _{m \rightarrow \infty}\left(\frac{m(2-\gamma) / 2-1}{\gamma m / 2+1}\right)=\frac{(2-\gamma) / 2}{\gamma / 2}=\frac{2-\gamma}{\gamma} \geq 2
$$

The last inequality follows since $\frac{2-\gamma}{\gamma}$ is a decreasing function, and for $\gamma \in\left(\frac{1}{3}, \frac{2}{3}\right]$ we have $\frac{2-\gamma}{\gamma} \in[2,5)$. Hence, for every $\alpha<2$ there exists an $m_{b}$ such that $\frac{m_{b}(2-\gamma) / 2-1}{\gamma m_{b} / 2+1}>\alpha$.

- $\frac{2 m}{3}<d\left(c_{1}, c_{2}\right) \leq m$ : Because $m$ is assumed to be significantly large and because the points of $\mathcal{C}_{1}, \mathcal{C}_{2}$ alternate in the metric cycle, we can find a point $j \in \mathcal{C}_{2}$ in the shortest path between $c_{1}$ and $c_{2}$, which will be approximately in the middle of the path. Letting $\gamma \in\left(\frac{2}{3}, 1\right]$ such that $d\left(c_{1}, c_{2}\right)=\gamma m$, we have $\frac{\gamma m}{2}-1 \leq d\left(j, c_{1}\right), d\left(j, c_{2}\right) \leq \frac{\gamma m}{2}+1$ (in Figure 2.1 we might have $c_{1}=j_{2}, c_{2}=j_{10}$ and $\left.j=j_{14}, \pi(j)=j_{10}\right)$. Consider now $\pi(j)$, and without loss of generality assume that $d\left(\pi(j), c_{1}\right) \geq d\left(\pi(j), c_{2}\right)$ (when $d\left(\pi(j), c_{1}\right) \leq d\left(\pi(j), c_{2}\right)$ the situation is symmetric, with the roles of $c_{1}, c_{2}$ switched.).

At first, suppose that $\pi(j)$ is a point in the shortest path between $c_{1}$ and $c_{2}$ (in the example of Figure $2.1 c_{1}=j_{2}, c_{2}=j_{10}$ and $j=j_{14}$ would result in that). Thus, because $d(j, \pi(j))=$ $m / 2, d\left(\pi(j), c_{1}\right) \geq d\left(\pi(j), c_{2}\right)$ and $d\left(c_{1}, c_{2}\right) \leq m$, we can focus on the line segment $c_{1}, j, \pi(j), c_{2}$, where the triangle inequality holds with equality. Here we get,

$$
d\left(\pi(j), c_{2}\right)=d\left(j, c_{2}\right)-d(j, \pi(j)) \leq \frac{\gamma m}{2}+1-\frac{m}{2}=\frac{(\gamma-1) m}{2}+1 \leq 1
$$

In addition,

$$
d\left(\pi(j), c_{1}\right)=d(j, \pi(j))+d\left(j, c_{1}\right) \geq \frac{m}{2}+\frac{\gamma m}{2}-1=\frac{(1+\gamma) m}{2}-1
$$

The second case we consider is when $\pi(j)$ is not on the shortest path between $c_{1}$ and $c_{2}$ (in Figure 2.1 take for instance $c_{1}=j_{1}, c_{2}=j_{11}$ and hence $j=j_{14}$ and $\pi(j)=j_{10}$ ). In that scenario, because $d\left(\pi(j), c_{1}\right) \geq d\left(\pi(j), c_{2}\right)$, we turn our attention to the line segment $c_{1}, j, c_{2}, \pi(j)$, where the triangle inequality holds with equality. Here we have

$$
\begin{aligned}
& d\left(\pi(j), c_{2}\right)=d(j, \pi(j))-d\left(j, c_{2}\right) \leq \frac{m}{2}-\frac{\gamma m}{2}+1=\frac{(1-\gamma) m}{2}+1 \\
& d\left(\pi(j), c_{1}\right)=d(j, \pi(j))+d\left(j, c_{1}\right) \geq \frac{m}{2}+\frac{\gamma m}{2}-1=\frac{(1+\gamma) m}{2}-1
\end{aligned}
$$

Therefore, in every case we have the following:

$$
\begin{equation*}
d\left(\pi(j), c_{1}\right) \geq \frac{(1+\gamma) m}{2}-1 \text { and } d\left(\pi(j), c_{2}\right) \leq \frac{(1-\gamma) m}{2}+1 \tag{2.26}
\end{equation*}
$$

Now that we have the bounds (2.26) for the assignment distance of $\pi(j)$ to both centers, we proceed with the final case analysis.

Suppose that $\pi(j)$ gets assigned to $c_{1}$. Then from $\pi(j)$ 's perspective, the best possible situation is if its own assignment distance is exactly $\frac{(1+\gamma) m}{2}-1$, and $j$ gets an assignment distance of $\frac{\gamma m}{2}+1$. In this case, the ratio $\frac{(1+\gamma) m / 2-1}{\gamma m / 2+1}$ is an increasing function of $m$, because $(1+\gamma) / 2, \gamma / 2>0$. In addition we have:

$$
\lim _{m \rightarrow \infty} \frac{(1+\gamma) m / 2-1}{\gamma m / 2+1}=\frac{1+\gamma}{\gamma} \geq 2
$$

The last inequality is because $\frac{1+\gamma}{\gamma}$ is a decreasing function and $\gamma \leq 1$. Hence, for every $\alpha<2$, there exists an $m_{c}$ such that $\frac{(1+\gamma) m_{c} / 2-1}{\gamma m_{c} / 2+1}>\alpha$. Thus, even in the ideal situation for $\pi(j)$, if $m$ is larger than $m_{c}$ its fairness constraint for $\alpha<2$ will be unsatisfiable.

On the other hand, suppose that $\pi(j)$ gets assigned to $c_{2}$. Then from $j$ 's perspective, the best possible situation is if it gets an assignment distance of $\frac{\gamma m}{2}-1$, and $\pi(j)$ has assignment distance exactly $\frac{(1-\gamma) m}{2}+1$. In this case, the ratio $\frac{\gamma m / 2-1}{(1-\gamma) m / 2+1}$ is an increasing function of $m$, because $(1-\gamma) / 2, \gamma / 2>0$. Also:

$$
\lim _{m \rightarrow \infty} \frac{\gamma m / 2-1}{(1-\gamma) m / 2+1}=\frac{\gamma}{1-\gamma}>2
$$

The last inequality is because $\frac{\gamma}{1-\gamma}$ is an increasing function and $\gamma>2 / 3$. Hence, for every $\alpha<2$, there exists an $m_{d}$ such that $\frac{\gamma m_{d} / 2-1}{(1-\gamma) m_{d} / 2+1}>\alpha$. Thus, even in the ideal situation for $j$, if $m$ is larger than $m_{d}, j$ 's fairness constraint for $\alpha<2$ will be unsatisfiable.

The analysis is exhaustive, because the maximum distance between two points in the metric is
$m$. Further, we see that if we set $m=4 \max \left\{m_{a}, m_{b}, m_{c}, m_{d}\right\}$, then in every possible scenario there will exist a point whose fairness constraint for $\alpha<2$ will not be satisfiable.

Moving on, we show that for $\alpha \geq 2$ there is always a feasible solution to both our problems, and hence we settle the important question of what is the smallest value of $\alpha$ for which EQCENTER-PP and EQCENTER-AG are well-defined.

Lemma 2.2.2. Consider a set of points $\mathcal{C}$ in a metric space with distance function $d$, where $|\mathcal{C}| \geq 2$. Then there exists an efficient way of finding two distinct points $c_{1}, c_{2} \in \mathcal{C}$ and an assignment $\phi: \mathcal{C} \mapsto\left\{c_{1}, c_{2}\right\}$, such that for every $j \in \mathcal{C}$ we have $\frac{d\left(c_{1}, c_{2}\right)}{2} \leq d(j, \phi(j)) \leq d\left(c_{1}, c_{2}\right)$.

Proof. At first, choose $c_{1}, c_{2}$ to be the two points of $\mathcal{C}$ that are the furthest apart, i.e. $\left(c_{1}, c_{2}\right)=$ $\arg \max _{x, y \in \mathcal{C}} d(x, y)$. Then, for every $j \in \mathcal{C}$ set $\phi(j)=\arg \max _{c \in\left\{c_{1}, c_{2}\right\}} d(j, c)$. In other words, given the chosen centers, each point is assigned to the center that is furthest from it in the metric. Let also $\bar{\phi}(j)$ be the center to which $j$ is not assigned to. For any $j \in \mathcal{C}$, combining the triangle inequality and the fact that $d(j, \bar{\phi}(j)) \leq d(j, \phi(j))$, will give us:

$$
d\left(c_{1}, c_{2}\right) \leq d(j, \phi(j))+d(j, \bar{\phi}(j)) \leq 2 d(j, \phi(j)) \Longrightarrow d\left(c_{1}, c_{2}\right) / 2 \leq d(j, \phi(j))
$$

Finally, by the way we chose $c_{1}$ and $c_{2}$ we also get $d(j, \phi(j)) \leq d\left(c_{1}, c_{2}\right)$.

Theorem 2.2.3. For both EQCENTER-PP and EQCENTER-AG, every instance with $\alpha \geq 2$ admits a feasible solution.

Proof. Suppose that as an instance to either problem we are given a set of points $\mathcal{C}$ together with their associated similarity sets $\mathcal{S}_{j}, k \geq 2$ and $\alpha \geq 2$. Since $k \geq 2$, we can use Lemma 2.2.2 and get a set of two centers $\left\{c_{1}, c_{2}\right\}$ and an assignment function $\phi: \mathcal{C} \mapsto\left\{c_{1}, c_{2}\right\}$, such that for all


Figure 2.2: PoF example: Lines represent distances, such that $d\left(j_{1}, j_{2}\right)=R^{\prime}, d\left(j_{2}, j_{3}\right)=R$.
$j \in \mathcal{C}$ we have $d\left(c_{1}, c_{2}\right) / 2 \leq d(j, \phi(j)) \leq d\left(c_{1}, c_{2}\right)$. In the case of constraint (1.1), for every $j \in \mathcal{C}$ and any $j^{\prime} \in \mathcal{S}_{j}$ we have $d(j, \phi(j)) \leq d\left(c_{1}, c_{2}\right) \leq 2 d\left(j^{\prime}, \phi\left(j^{\prime}\right)\right) \leq \alpha d\left(j^{\prime}, \phi\left(j^{\prime}\right)\right)$. Furthermore, since any feasible solution for constraint (1.1) is also a feasible solution for constraint (1.2), the proof is concluded.

Another structural notion that interests us, is that of the Price of Fairness (PoF). For a given instance of either of our problems, PoF is the ratio of the value of the optimal solution to the problem, over the the optimal unfair value. The latter is defined as the optimal value of the given instance, when we drop the fairness constraint and simply solve $k$-center. As is the case in most fair clustering literature, we show that in general PoF can be arbitrarily large.

## Theorem 2.2.4. Both EQCEnter-PP and EQCenter-AG have instances with unbounded PoF.

Proof. We will use the example of Figure 2.2 for both problems. Consider three points $j_{1}, j_{2}, j_{3}$ on the line, where $d\left(j_{1}, j_{2}\right)=R^{\prime}, d\left(j_{2}, j_{3}\right)=R$ and $d\left(j_{1}, j_{3}\right)=R^{\prime}+R$. Moreover, let $\alpha=2, k=$ $2, \mathcal{S}_{j_{1}}=\left\{j_{2}\right\}, \mathcal{S}_{j_{2}}=\left\{j_{1}, j_{3}\right\}, \mathcal{S}_{j_{3}}=\left\{j_{2}\right\}$, and assume that $R^{\prime} \ll R$ as well as $\frac{R}{R^{\prime}} \gg 2$.

First, note that in the absence of the fairness constraints, the optimal solution for $k$-center occurs when $j_{2}$ and $j_{3}$ (or $j_{1}$ and $j_{3}$ ) are chosen as centers, and its corresponding value is $R^{\prime}$.

Moving forward, we show that the optimal solution for the fair variants has value at least $R$ (note that the existence of such a solution is guaranteed by Theorem 2.2.3). This implies that PoF is at least $\frac{R}{R^{\prime}}$, and since $R^{\prime} \ll R$ this can be arbitrarily large. We proceed with a case analysis.

Initially, consider a solution that uses only one center. If that center is either $j_{1}$ or $j_{3}$, then regardless of feasibility issues the value of the corresponding solution will be $R^{\prime}+R$. On the other hand, if $j_{2}$ is chosen as a center, then we will necessarily have $\phi\left(j_{1}\right)=\phi\left(j_{2}\right)=\phi\left(j_{3}\right)=j_{2}$ as the only possible assignment. Hence, because $d\left(j_{2}, \phi\left(j_{2}\right)\right)=0$ and $\mathcal{S}_{j_{1}}=\left\{j_{2}\right\}$, no matter what constraint we have for $j_{1}$, it cannot be satisfied. Thus, $j_{2}$ can never be a center on its own. Now we consider solutions that use exactly two centers.

- Let $\left\{j_{1}, j_{2}\right\}$ be the set of chosen centers. In this case note that $d\left(j_{2}, \phi\left(j_{2}\right)\right) \in\left\{0, R^{\prime}\right\}$. If $d\left(j_{2}, \phi\left(j_{2}\right)\right)=0$, then clearly both constraints (1.1) and (1.2) for $j_{3}$ cannot be satisfied. If $d\left(j_{2}, \phi\left(j_{2}\right)\right)=R^{\prime}$, then $\frac{d\left(j_{3}, \phi\left(j_{3}\right)\right)}{d\left(j_{2}, \phi\left(j_{2}\right)\right)} \geq \frac{R}{R^{\prime}}>2$, and hence again the fairness constraints for $j_{3}$ cannot be satisfied. Therefore, $\left\{j_{1}, j_{2}\right\}$ will never be the chosen set of centers.
- Let $\left\{j_{2}, j_{3}\right\}$ be the set of chosen centers. If we want to satisfy the fairness constraints for $j_{1}$, we should set $\phi\left(j_{2}\right)=j_{3}$, because otherwise the assignment distance of $j_{2}$ will be 0 . Having $\phi\left(j_{2}\right)=j_{3}$ immediately implies that if there is a feasible solution for this set of centers, its value should be at least $d\left(j_{2}, \phi\left(j_{2}\right)\right)=R$.
- Let $\left\{j_{1}, j_{3}\right\}$ be the set of chosen centers. To begin with, see that an assignment that leads to a value of $R^{\prime}$ is not possible. The only mapping that leads to such a solution is $\phi\left(j_{1}\right)=$ $j_{1}, \phi\left(j_{2}\right)=j_{1}, \phi\left(j_{3}\right)=j_{3}$. However, this violates both constraints (1.1) and (1.2) for $j_{2}$, since all points in $\mathcal{S}_{j_{2}}$ will have an assignment distance of 0 . Thus, because of the discrete values of the metric space, any feasible solution using $\left\{j_{1}, j_{3}\right\}$ as its centers should have value at least $R$.


### 2.2.2 Approximation Algorithms for EQCenter-PP and EQCenter-AG

Suppose that we are given an instance of EQCENTER with $\alpha, k \geq 2$, and we are either solving EQCENTER-PP or EQCENTER-AG. In addition, let $R_{m}=\max _{j \in \mathcal{C}} R_{j}$ and let $R^{*}$ denote the value of the optimal solution of the corresponding problem.

In this section we provide a procedure that works under an explicitly given value $R$, with $R \geq R_{m}$. This process will either give a feasible solution $\left(S_{R}, \phi_{R}\right)$ with $\max _{j \in \mathcal{C}} d\left(j, \phi_{R}(j)\right) \leq$ $5 R$, or an infeasibility message. The latter message indicates with absolute certainty that $R<R^{*}$.

The aforementioned procedure suffices to guarantee the result of Theorem 1.4.6. Because $R^{*}$ is always the distance between two points in $\mathcal{C}$, the total number of possible values for it is only polynomial, specifically at most $\binom{|\mathcal{C}|}{2}$. Hence, we can run the procedure for all such distances that are at least $R_{m}$, and in the end keep $\left(S_{R}, \phi_{R}\right)$ for the minimum guess $R$ for which we did not receive an infeasibility message. If $R_{m} \leq R^{*}$, then our returned solution is guaranteed to have value at most $5 R^{*}$, because $R^{*}$ is one of the target values we tested. On the other hand, when $R_{m}>R^{*}$, the iteration with $R_{m}$ as the guess cannot return and infeasibility message, and thus it provide a solution of value at most $5 R_{m}$. As a side note, we can speed up the runtime of this approach by using binary search over the guesses $R$, instead of a naive brute-force method.

Therefore, apart from the input instance, assume that we are also given a target value $R$ with $R \geq R_{m}$. Our framework begins by choosing an appropriate set of centers $S$. The full details of this step are presented in Algorithm 4. Besides choosing this set $S$, Algorithm 4 also creates a partition $P_{1}, P_{2}, \ldots P_{T}$ of $S$ for some $T \leq|\mathcal{C}|$, and returns sets $G_{c} \subseteq \mathcal{C}$ for every $c \in S$.

Initially, all point of $\mathcal{C}$ are considered uncovered $(U=\mathcal{C})$. The algorithm works by trying to expand the current set of centers $P_{t}$ as much as possible, via finding a new center that is currently

```
Algorithm 4: Choosing an initial set of centers
    \(S \leftarrow \emptyset, U \leftarrow \mathcal{C}, P_{0} \leftarrow \emptyset, t \leftarrow 0 ;\)
    while \(U \neq \emptyset\) do
        \(Q \leftarrow\left\{c \in U \mid \exists c^{\prime} \in P_{t}\right.\) such that \(\left.d\left(c, c^{\prime}\right) \leq 3 R\right\} ;\)
        if \(Q \neq \emptyset\) then
            Choose a point \(c \in Q\);
            \(P_{t} \leftarrow P_{t} \cup\{c\} ;\)
            \(S \leftarrow S \cup\{c\} ;\)
            \(G_{c} \leftarrow\{j \in U \mid d(j, c) \leq 2 R\} ;\)
            \(U \leftarrow U \backslash G_{c} ;\)
        else
            Choose an arbitrary \(c \in U\);
            \(t \leftarrow t+1\);
            \(P_{t} \leftarrow\{c\} ;\)
            \(S \leftarrow S \cup\{c\} ;\)
            \(G_{c} \leftarrow\{j \in U \mid d(j, c) \leq 2 R\} ;\)
            \(U \leftarrow U \backslash G_{c} ;\)
        end
    end
    Return the set \(S\), the partition \(P_{1}, P_{2}, \ldots P_{t}\) of \(S\), and the sets \(G_{c}\) for every \(c \in S\);
```

uncovered and is within distance $3 R$ from some center already placed in $P_{t}$. If no such point exists, then we never deal with $P_{t}$ again, and we move on to create $P_{t+1}$ by choosing an arbitrary uncovered point as the first center for it. In additional, every time a center $c$ is chosen, it covers all uncovered points that are within distance $2 R$ from it, and these points constitute the set $G_{c}$. This process is repeated until all points get covered, i.e., until the set $U$ becomes empty.

For every $c \in S$, let $t(c)$ be the index of the partition set $c$ belongs to, i.e., $c \in P_{t(c)}$. We also define $S_{I}=\left\{c \in S:\left|P_{t(c)}\right|=1\right\}$ and $S_{N}=S \backslash S_{I}$. We interpret the centers of $S_{I}$ as being isolated, since for each $c \in S_{I}$ its corresponding partition set contains only $c$, i.e., $P_{t(c)}=\{c\}$. On the other hand, the centers of $S_{N}$ are non-isolated, in the sense of having $\left|P_{t(c)}\right|>1$ for each $c \in S_{N}$. In addition, for every point $j \in \mathcal{C}$, let $\rho(j)$ the center of $S$ that covered $j$, i.e., $j \in G_{\rho(j)}$. Note that $d(j, \rho(j)) \leq 2 R$. Finally, let $\mathcal{C}_{I}=\left\{j \in \mathcal{C}: \rho(j) \in S_{I}\right\}$ and $\mathcal{C}_{N}=\mathcal{C} \backslash \mathcal{C}_{I}$, where $\mathcal{C}_{I}$ are
the points that got covered by isolated centers, and $\mathcal{C}_{N}$ the points covered by non-isolated centers.

Observation 2.2.5. For every distinct $c, c^{\prime} \in S$ we have $d\left(c, c^{\prime}\right)>2 R$.

Observation 2.2.6. For every $c \in S_{N}$, there exists a different $c^{\prime} \in S_{N}$ with $d\left(c, c^{\prime}\right) \leq 3 R$.

Observation 2.2.7. The sets $G_{c}$ for all $c \in S$, induce a partition of $\mathcal{C}$.

The previous observations follow trivially from the definition of Algorithm 4. However, Observation 2.2.6 is of particular importance, since it will allow us to carefully control the assignment distances of points later on, in a way that would satisfy the underlying fairness constraints.

Lemma 2.2.8. For any $c \in S_{I}$, we have $d\left(j, j^{\prime}\right)>R$ for all $j \in G_{c}$ and all $j^{\prime} \in \mathcal{C} \backslash G_{c}$.

Proof. Focus on such a $c \in S_{I}$, and for the sake of contradiction assume that there exists a $j \in G_{c}$ and a $j^{\prime} \in \mathcal{C} \backslash G_{c}$ for which $d\left(j, j^{\prime}\right) \leq R$. Let $c^{\prime} \neq c$ the center of $S$ with $c^{\prime}=\rho\left(j^{\prime}\right)$.

At first, suppose that during the execution of Algorithm $4 c$ entered $S$ before $c^{\prime}$. Having $\left|P_{t(c)}\right|=1$ means that when $P_{t(c)}=\{c\}$, the algorithm tried to find a point in $U$ within distance $3 R$ from $c$ but failed. However, at that time $j^{\prime}$ was still in $U$, because $j^{\prime} \in G_{c^{\prime}}$ and $c^{\prime}$ entered $S$ after $c$. In addition $d\left(j^{\prime}, c\right) \leq d\left(j, j^{\prime}\right)+d(j, c) \leq 3 R$, and thus we reached a contradiction.

Now assume that $c^{\prime}$ entered $S$ before $c$. This implies that $t\left(c^{\prime}\right)<t(c)$, because $\left|P_{t(c)}\right|=1$. When the algorithm stopped expanding $P_{t\left(c^{\prime}\right)}$, there was not any point of $U$ within distance $3 R$ from a center of $P_{t\left(c^{\prime}\right)}$. However, at that moment $j$ was still in $U$, because $j \in G_{c}$ and $t\left(c^{\prime}\right)<t(c)$. In addition $d\left(j, c^{\prime}\right) \leq d\left(j, j^{\prime}\right)+d\left(j^{\prime}, c^{\prime}\right) \leq 3 R$, and so we once again reach a contradiction.

By using Lemma 2.2.8 and the fact that $R \geq R_{m}$, we immediately get the following.

Corollary 2.2.9. For every $c \in S_{I}$, we have $\mathcal{S}_{j} \subseteq G_{c} \subseteq \mathcal{C}_{I}$ for all $j \in G_{c}$.

Corollary 2.2.10. For every $j \in \mathcal{C}_{N}$, we have $\mathcal{S}_{j} \cap \mathcal{C}_{I}=\emptyset$.

In words, Corollary 2.2 .9 says that the similarity set of a point $j \in \mathcal{C}_{I}$ is completely contained in $G_{\rho(j)}$, where of course $\rho(j) \in S_{I}$ and $G_{\rho(j)} \subseteq \mathcal{C}_{I}$. Similarly, Corollary 2.2.10 says that the similarity set of a point $j \in \mathcal{C}_{N}$ is completely contained in $\mathcal{C}_{N}$.

After computing the set of centers $S$, our approach proceeds by constructing the appropriate assignment function. This will occur in two steps. The first step takes care of the points in $\mathcal{C}_{I}$, by choosing a new set of centers $S_{I}^{\prime} \subseteq \mathcal{C}_{I}$, and by constructing an assignment $\phi_{I}: \mathcal{C}_{I} \mapsto S_{I}^{\prime}$. The second step handles the points of $\mathcal{C}_{N}$ via a mapping $\phi_{N}: \mathcal{C}_{N} \mapsto S_{N}$. This is well-defined, since $\mathcal{C}_{I} \cap \mathcal{C}_{N}=\emptyset$. Note now that due to Corollary 2.2.9, the fairness constraint of a point $j \in \mathcal{C}_{I}$ is only affected by $\phi_{I}$, since $\mathcal{S}_{j} \subseteq G_{\rho(j)} \subseteq \mathcal{C}_{I}$ and $\mathcal{C}_{I} \cap \mathcal{C}_{N}=\emptyset$. Similarly, due to Corollary 2.2.10, the fairness constraint of a $j \in \mathcal{C}_{N}$ is only affected by $\phi_{N}$, since $\mathcal{S}_{j} \subseteq \mathcal{C}_{N}$ and $\mathcal{C}_{I} \cap \mathcal{C}_{N}=\emptyset$. Thus, we can study the satisfaction of fairness constraints separately on $\mathcal{C}_{I}$ for $\phi_{I}$, and on $\mathcal{C}_{N}$ for $\phi_{N}$.

Algorithm 5 demonstrates the details of the first assignment step. The algorithm operates by trying to "guess" if the optimal solution uses exactly one center inside each $G_{c}$ for $c \in S_{I}$. If it does, so will our algorithm. If not, then our approach will open exactly two centers, and will subsequently construct an assignment that will satisfy the appropriate fairness constraint.

Lemma 2.2.11. After the execution of Algorithm 5, for every $j \in \mathcal{C}_{I}$ we have that the constructed assignment $\phi_{I}$ will 1) satisfy j's fairness constraint, and 2) guarantee $d\left(j, \phi_{I}(j)\right) \leq 4 R$.

Proof. At first, due to Observation 2.2.7, Algorithm 5 sets the value $\phi_{I}(j)$ for each $j \in \mathcal{C}_{I}$ exactly once. In addition, we know that for every $j \in \mathcal{C}_{I}$, all points of $\mathcal{S}_{j}$ will have their assignment set in the same iteration of Algorithm 5, since $\rho(j) \in S_{I}$ and by Corollary 2.2.9 we have $\mathcal{S}_{j} \subseteq G_{\rho(j)}$. For a point $j \in \mathcal{C}_{I}$, when $\rho(j)$ is considered by Algorithm 5 there are two possible scenarios.

```
Algorithm 5: Assignment for the points of \(\mathcal{C}_{I}\)
    \(S_{I}^{\prime} \leftarrow \emptyset ;\)
    for every \(c \in S_{I}\) do
            Check if there exists any \(j \in G_{c}\), such that assigning all points of \(G_{c}\) to \(j\) would
                result in the appropriate fairness constraint being satisfied for each \(j^{\prime} \in G_{c}\). Note
                that checking this feasibility condition is possible due to Corollary 2.2.9. If there
                existed such a \(j\), set \(S_{I}^{\prime} \leftarrow S_{I}^{\prime} \cup\{j\}\) and \(\phi_{I}\left(j^{\prime}\right) \leftarrow j\) for all \(j^{\prime} \in G_{c}\);
            If you could not find such a \(j\), use the algorithm of Lemma 2.2.2 on the points of
                \(G_{c}\). This will return two points \(c_{1}, c_{2} \in G_{c}\) and an assignment \(\phi: G_{c} \mapsto\left\{c_{1}, c_{2}\right\}\).
                Then set \(S_{I}^{\prime} \leftarrow S_{I}^{\prime} \cup\left\{c_{1}, c_{2}\right\}\) and \(\phi_{I}\left(j^{\prime}\right) \leftarrow \phi\left(j^{\prime}\right)\) for all \(j^{\prime} \in G_{c}\);
    end
    Return \(S_{I}^{\prime}\) and \(\phi_{I}\);
```

In the first we have $\left|S_{I}^{\prime} \cap G_{\rho(j)}\right|=1$. If that happens, all points of $G_{\rho(j)}$ are assigned to the only point of $S_{I}^{\prime} \cap G_{\rho(j)}$, and we are also sure that the fairness constraint of all of them is satisfied. Otherwise, we have $\left|S_{I}^{\prime} \cap G_{\rho(j)}\right|=2$, as a result of running the algorithm of Lemma 2.2.2 on $G_{\rho(j)}$. By using the assignment guarantees of that algorithm, it is easy to see that the fairness constraints for all $j^{\prime} \in G_{\rho(j)}$ will again be satisfied. Hence, in both cases the corresponding fairness constraint is satisfied for $j$.

Thus, $d\left(j, \phi_{I}(j)\right) \leq d(j, \rho(j))+d\left(\phi_{I}(j), \rho(j)\right) \leq 4 R$, since $\phi_{I}(j) \in G_{\rho(j)}$ in each case.

Lemma 2.2.12. If $R \geq R^{*}$, then after the execution of Algorithm 5 we have $\left|S_{I}^{\prime}\right|+\left|S_{N}\right| \leq k$.

Proof. Let $S^{*}$ be the optimal set of centers, and $\phi^{*}$ the corresponding optimal assignment. The following two statements rely on the fact that $R \geq R^{*}$. First, by Observation 2.2 .5 note that for two distinct points $c, c^{\prime} \in S_{N}$ we must have $\phi^{*}(c) \neq \phi^{*}\left(c^{\prime}\right)$. Second, due to Lemma 2.2.8 we also have $\phi^{*}(c) \notin \mathcal{C}_{I}$ for every $c \in S_{N}$. The two previous statements imply $\left|S_{N}\right| \leq\left|S^{*} \backslash \mathcal{C}_{I}\right|$.

Now focus on $S^{*} \cap \mathcal{C}_{I}$, and see that $\left|S^{*} \cap \mathcal{C}_{I}\right|=\sum_{c \in S_{I}}\left|S^{*} \cap G_{c}\right|$ due to Observation 2.2.7 and the definition of $\mathcal{C}_{I}$. Further, due to Lemma 2.2.8 and the fact that $R \geq R^{*}$, we have that $\left|S^{*} \cap G_{c}\right| \geq 1$ for every $c \in S_{I}$. If $\left|S^{*} \cap G_{c}\right|=1$, then we know that the optimal solution assigns
all points of $G_{c}$ to the unique point of $S^{*} \cap G_{c}$. This assignment is obviously feasible, and thus the first part of Algorithm 5 can identify it and guarantee $\left|S_{I}^{\prime} \cap G_{c}\right|=1$. Otherwise, if $\left|S^{*} \cap G_{c}\right| \geq 2$, then Algorithm 5 ensures that $\left|S_{I}^{\prime} \cap G_{c}\right| \leq 2$. Therefore, we get

$$
\left|S_{I}^{\prime}\right|=\sum_{c \in S_{I}}\left|S_{I}^{\prime} \cap G_{c}\right| \leq \sum_{c \in S_{I}}\left|S^{*} \cap G_{c}\right|=\left|S^{*} \cap \mathcal{C}_{I}\right|
$$

Putting everything together yields

$$
\left|S_{I}^{\prime}\right|+\left|S_{N}\right| \leq\left|S^{*} \cap \mathcal{C}_{I}\right|+\left|S^{*} \backslash \mathcal{C}_{I}\right|=\left|S^{*}\right| \leq k
$$

Using the contrapositive of Lemma 2.2.12, we see that if $\left|S_{I}^{\prime}\right|+\left|S_{N}\right|>k$ then $R<R^{*}$, and hence we can safely return as our answer an infeasibility message.

Before we proceed to the second step of our assignment process, we need some extra notation. For each $c \in S_{N}$ define $H_{c}^{1}=\left\{j \in \mathcal{C}_{N} \mid d(j, c) \leq R\right\}$ and $H_{c}^{2}=G_{c} \backslash\left(\bigcup_{c^{\prime} \in S_{N}} H_{c^{\prime}}^{1}\right)$. Combining Observation 2.2.5, Observation 2.2.7 and the way we constructed the sets $H_{c}^{1}, H_{c}^{2}$, it is easy to see that for each $j \in \mathcal{C}_{N}$ exactly one of the following two cases will hold.

- The point $j$ belongs to exactly one $H_{c}^{1}$ for some $c \in S_{N}$. In addition, $j$ clearly does not belong to any set $H_{c^{\prime}}^{2}$ for $c^{\prime} \in S_{N}$. In this case, we call $j$ a type-1 point, and we set $\pi(j)=c$.
- The point $j$ belongs to $H_{\rho(j)}^{2}$. In addition, $j$ does not belong to any $H_{c}^{1}$ for $c \in S_{N}$, and it does not belong to any $H_{c}^{2}$ with $c \neq \rho(j)$. Here we call $j$ a type-2 point, and set $\pi(j)=\rho(j)$.

Further, let $\mathcal{C}_{N}^{1}=\left\{j \in \mathcal{C}_{N} \mid j\right.$ is a type-1 point $\}$ and $\mathcal{C}_{N}^{2}=\left\{j \in \mathcal{C}_{N} \mid j\right.$ is a type-2 point $\}$. Therefore, $\mathcal{C}_{N}^{1} \cap \mathcal{C}_{N}^{2}=\emptyset$ and $\mathcal{C}_{N}^{1} \cup \mathcal{C}_{N}^{2}=\mathcal{C}_{N}$. Finally, the definition of a type- 2 point implies:

```
Algorithm 6: Assignment for the points of \(\mathcal{C}_{N}\)
    for every \(j \in \mathcal{C}_{N}\) do
        if \(j \in \mathcal{C}_{N}^{1}\) then
            \(\phi_{N}(j) \leftarrow \arg \min _{c \in\left(S_{N} \backslash\{\pi(j)\}\right)} d(j, c) ; \quad / /\) Case (A)
        end
        if \(j \in \mathcal{C}_{N}^{2}\) then
            if \(\exists c \in\left(S_{N} \backslash\{\pi(j)\}\right): d(j, c) \leq 2 R\) then
                \(\phi_{N}(j) \leftarrow \arg \max _{c^{\prime} \in S_{N}: d\left(j, c^{\prime}\right) \leq 2 R} d\left(j, c^{\prime}\right) ; \quad / /\) Case (B)
            else
                \(\phi_{N}(j) \leftarrow \arg \min _{c^{\prime} \in\left(S_{N} \backslash\{\pi(j)\}\right)} d\left(j, c^{\prime}\right) ; \quad / /\) Case (C)
            end
        end
    end
    Return the assignment \(\phi_{N}: \mathcal{C}_{N} \mapsto S_{N}\);
```

Observation 2.2.13. For all $j \in \mathcal{C}_{N}^{2}$, we have $d(j, \pi(j)) \leq 2 R$ and $d(j, c)>R$ for all $c \in S_{N}$.

The distinction between type-1 and type-2 points is necessary for satisfying the fairness constraints. Notice that by construction of $S_{N}$ type-1 points are more "privilleged", since they have an available center within distance at most $R$ from them. On the other hand, type-2 points do not have such an advantage. Thus, the assignment process should be aware of this discrepancy, so it can favor type-2 points in a controlled way that will satisfy everyone's fairness constraint.

Algorithm 6 demonstrates the full details of constructing the assignment $\phi_{N}: \mathcal{C}_{N} \mapsto S_{N}$. The high-level intuition behind it follows. At first, we try to provide each point $j$ with an assignment distance in the range $[R, 5 R]$, something that is possible due to Observation 2.2.6. However, since $\alpha$ might be less than 5 , we are very careful in how we handle the assignment of similar points. The latter is achieved by considering type-1 and type- 2 points independently, in a manner that is aware of where the potential similar points of each type may be.

Lemma 2.2.14. For any point $j \in \mathcal{C}_{N}$ we have:

- $d(j, \pi(j)) \leq R<d\left(j, \phi_{N}(j)\right) \leq 4 R$, if $j$ gets assigned to $\phi_{N}(j)$ through Case (A).
- $R<d(j, \pi(j)) \leq d\left(j, \phi_{N}(j)\right) \leq 2 R$, if $j$ gets assigned to $\phi_{N}(j)$ through Case (B).
- $d(j, \pi(j)) \leq 2 R<d\left(j, \phi_{N}(j)\right) \leq 5 R$, if $j$ gets assigned to $\phi_{N}(j)$ through Case (C).

Proof. In Case (A) $d(j, \pi(j)) \leq R$ since $j \in H_{\pi(j)}^{1}$. Also, due to Observation 2.2.5 there does not exist any point in $S_{N} \backslash\{\pi(j)\}$ that is within distance at most $R$ from $j$, and hence $d\left(j, \phi_{N}(j)\right)>$ $R \geq d(j, \pi(j))$. In addition, Observation 2.2.6 ensures that there exists a $c \in S_{N} \backslash\{\pi(j)\}$ such that $d(\pi(j), c) \leq 3 R$. Therefore, $d\left(j, \phi_{N}(j)\right) \leq d(j, c) \leq d(j, \pi(j))+d(\pi(j), c) \leq 4 R$.

The assignment guarantee for Case (B) follows trivially from Observation 2.2.13, and the way the algorithm operates in that situation.

In Case (C) we have $d(j, c)>2 R \geq d(j, \pi(j))$ for all $c \in S_{N} \backslash\{\pi(j)\}$. Also, Observation 2.2.6 ensures that there exists a $c^{\prime} \in S_{N} \backslash\{\pi(j)\}$ such that $d\left(\pi(j), c^{\prime}\right) \leq 3 R$. Hence

$$
d\left(j, \phi_{N}(j)\right) \leq d\left(j, c^{\prime}\right) \leq d(j, \pi(j))+d\left(\pi(j), c^{\prime}\right) \leq 5 R
$$

where $d(j, \pi(j)) \leq 2 R$ follows from Observation 2.2.13.

Lemma 2.2.14 immediately gives an upper bound of $5 R$ for the maximum assignment distance. However, it is the rest of the inequalities shown there that allow us to prove satisfaction of the fairness constraints by $\phi_{N}$. This is achieved in the following Lemma.

Lemma 2.2.15. For all $j \in \mathcal{C}_{N}$, we have $d\left(j, \phi_{N}(j)\right) \leq \alpha \cdot d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)$ for all $j^{\prime} \in \mathcal{S}_{j}$.

Proof. Consider some $j \in \mathcal{C}_{N}$ and some $j^{\prime} \in \mathcal{S}_{j}$. The proof of the statement will be based on an exhaustive case analysis. Before we proceed, we mention two inequalities that we will repeatedly use. At first, $d\left(j, j^{\prime}\right) \leq d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)$, because $d\left(j, j^{\prime}\right) \leq R_{m} \leq R$ and by Lemma 2.2.14 we have $d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)>R$. Moreover, $d\left(j^{\prime}, \pi\left(j^{\prime}\right)\right) \leq d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)$, again by using Lemma 2.2.14.

- Suppose that $j$ is a type- 1 point and $j^{\prime}$ is also a type- 1 point.

At first let $\pi(j) \neq \pi\left(j^{\prime}\right)$. Then $j$ can potentially be assigned to $\pi\left(j^{\prime}\right)$, and therefore we have $d\left(j, \phi_{N}(j)\right) \leq d\left(j, \pi\left(j^{\prime}\right)\right) \leq d\left(j, j^{\prime}\right)+d\left(j^{\prime}, \pi\left(j^{\prime}\right)\right) \leq 2 d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right) \leq \alpha \cdot d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)$.

Now let $\pi(j)=\pi\left(j^{\prime}\right)$. Because $j^{\prime}$ is a type-1 point and gets assigned according to Case (A), we know that $\phi_{N}\left(j^{\prime}\right) \neq \pi(j)$, Hence $j$ can potentially be assigned to $\phi_{N}\left(j^{\prime}\right)$. Therefore, $d\left(j, \phi_{N}(j)\right) \leq d\left(j, \phi_{N}\left(j^{\prime}\right)\right) \leq d\left(j, j^{\prime}\right)+d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right) \leq 2 d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right) \leq \alpha \cdot d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)$.

- Suppose that $j$ is a type- 1 point and $j^{\prime}$ is a type- 2 point.

At first assume $j^{\prime}$ received its assignment via Case (C). Then, by Lemma 2.2.14 we know that $d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)>2 R$. In addition, again by Lemma 2.2.14, we have $d\left(j, \phi_{N}(j)\right) \leq 4 R$. Thus, $d\left(j, \phi_{N}(j)\right) \leq 2 d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right) \leq \alpha \cdot d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)$.

Now assume that $j^{\prime}$ received its assignment through Case (B). Therefore, there exists $c \in$ $S \backslash\left\{\pi\left(j^{\prime}\right)\right\}$ with $d\left(j^{\prime}, c\right) \leq 2 R$. By the way Case (B) works and Observation 2.2.13, we also have $d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right) \geq \max \left(d\left(j^{\prime}, \pi\left(j^{\prime}\right)\right), d\left(j^{\prime}, c\right)\right)$. Let us now see what happens when $\pi\left(j^{\prime}\right)=\pi(j)$. Then $c \neq \pi(j)$, and thus $j$ can potentially be assigned to $c$. Therefore, $d\left(j, \phi_{N}(j)\right) \leq d(j, c) \leq d\left(j, j^{\prime}\right)+d\left(j^{\prime}, c\right) \leq d\left(j, j^{\prime}\right)+d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right) \leq 2 d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right) \leq$ $\alpha \cdot d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)$. On the other hand, if $\pi\left(j^{\prime}\right) \neq \pi(j)$, then $j$ can potentially get assigned to $\pi\left(j^{\prime}\right)$, and thus have $d\left(j, \phi_{N}(j)\right) \leq d\left(j, \pi\left(j^{\prime}\right)\right) \leq d\left(j, j^{\prime}\right)+d\left(j^{\prime}, \pi\left(j^{\prime}\right)\right) \leq \alpha \cdot d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)$.

- Suppose $j$ is a type-2 point, and also gets its assignment via Case (B). By Lemma 2.2.14 we have $d\left(j, \phi_{N}(j)\right) \leq 2 R$ and $d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)>R$. The statement follows since $\alpha \geq 2$.
- Suppose that $j$ is a type-2 point, $j^{\prime}$ is a type-1 point, and $j$ gets its assignment via Case (C).

At first, assume that $\phi_{N}\left(j^{\prime}\right) \neq \pi(j)$. In this case $j$ can potentially get assigned to $\phi_{N}\left(j^{\prime}\right)$,
and $d\left(j, \phi_{N}(j)\right) \leq d\left(j, \phi_{N}\left(j^{\prime}\right)\right) \leq d\left(j, j^{\prime}\right)+d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right) \leq \alpha \cdot d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)$.

Now assume that $\phi_{N}\left(j^{\prime}\right)=\pi(j)$. Because $j^{\prime}$ is a type-1 points and so $\phi_{N}\left(j^{\prime}\right) \neq \pi\left(j^{\prime}\right)$, we can infer that $\pi(j) \neq \pi\left(j^{\prime}\right)$. Also, $d\left(j, \pi\left(j^{\prime}\right)\right) \leq d\left(j, j^{\prime}\right)+d\left(j^{\prime}, \pi\left(j^{\prime}\right)\right) \leq 2 R$. However, the latter contradicts the assumption that $j$ got its assignment according to Case (C). Therefore, we know that $\phi_{N}\left(j^{\prime}\right) \neq \pi(j)$ necessarily.

- Suppose that both $j, j^{\prime}$ are type-2 points, and $j$ gets its assignment via Case (C).

At first, assume $\pi\left(j^{\prime}\right) \neq \pi(j)$. Then $j$ can potentially get assigned to $\pi\left(j^{\prime}\right)$, and therefore $d\left(j, \phi_{N}(j)\right) \leq d\left(j, \pi\left(j^{\prime}\right)\right) \leq d\left(j, j^{\prime}\right)+d\left(j^{\prime}, \pi\left(j^{\prime}\right)\right) \leq 2 d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right) \leq \alpha \cdot d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)$.

Now let $\pi\left(j^{\prime}\right)=\pi(j)$. To begin with, assume that there exists a $c \in S \backslash\{\pi(j)\}$ such that $d\left(j^{\prime}, c\right) \leq 2 R$. Moreover, because $c \neq \pi(j), j$ can potentially get assigned to $c$, and thus $d\left(j, \phi_{N}(j)\right) \leq d(j, c) \leq d\left(j, j^{\prime}\right)+d\left(j^{\prime}, c\right) \leq d\left(j, j^{\prime}\right)+d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right) \leq \alpha \cdot d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)$. To get $d\left(j^{\prime}, c\right) \leq d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right)$ we simply used the way Case (B) works. Finally, suppose that $\forall c \in S \backslash\{\pi(j)\}$ we have $d\left(j^{\prime}, c\right)>2 R$. Then $\phi_{N}\left(j^{\prime}\right) \neq \pi(j)$ and $j$ can get assigned to $\phi_{N}\left(j^{\prime}\right)$. Thus, $d\left(j, \phi_{N}(j)\right) \leq d\left(j, \phi_{N}\left(j^{\prime}\right)\right) \leq d\left(j, j^{\prime}\right)+d\left(j^{\prime}, \phi_{N}\left(j^{\prime}\right)\right) \leq \alpha \cdot d\left(j, \phi_{N}\left(j^{\prime}\right)\right)$

Combining Lemmas 2.2.14 and 2.2.15 we get the following.

Lemma 2.2.16. After the execution of Algorithm 6, for every $j \in \mathcal{C}_{N}$ we have that the constructed assignment $\phi_{N}$ will 1) satisfy j's fairness constraint, and 2) guarantee $d\left(j, \phi_{N}(j)\right) \leq 5 R$.

Combining Lemmas 2.2.11, 2.2.16 and 2.2.12 with the fact that the number of centers used is $\left|S_{I}^{\prime}\right|+\left|S_{N}\right|$, we see that we provide a procedure that for a guess $R \geq R_{m}$ works as follows. It either returns a solution with maximum assignment distance $5 R$, or returns an infeasibility message indicating $R<R^{*}$. As mentioned earlier, this concludes the proof of Theorem 1.4.6.

### 2.2.2.1 Cases with Bounded PoF

As we have already shown in Theorem 2.2.4, the Price of Fairness for both variants of EQCENTER we consider, can in general be unbounded. However, we are going to demonstrate that when the similarity sets $\mathcal{S}_{j}$ satisfy certain properties, slight modifications to our algorithms can provide guarantees with bounded PoF. Specifically, the objective function value provided by the algorithm will be comparable to the optimal unfair value, up to some constant factor.

For the given instance of EQCENTER, let $R_{u n f}^{*}$ be the value of the optimal $k$-center solution for this instance. Alternatively, to define $R_{u n f}^{*}$ we drop the fairness constraints from the model's requirements, and simply focus on the optimal value of the underlying vanilla clustering problem.

The first scenario we study is when for all $j \in \mathcal{C}$ we have $R_{j} \leq R_{u n f}^{*}$, and consequently $R_{m} \leq R_{u n f}^{*}$. Consider now the following modification to our main algorithm, which consists of only changing Algorithm 5, and thus the construction of $S_{I}^{\prime}$ and $\phi_{I}$. If for some $c \in S_{I}$ we have $\left|G_{c}\right|=1$, then we use $c$ as a center and set $\phi_{I}(c)=c$. If for some $c \in S_{I}$ we have $\left|G_{c}\right| \geq 2$, then we immediately use the procedure of Lemma 2.2.2 in order to choose centers, without checking if only one point of $G_{c}$ can yield a feasible solution. This modification yields Theorem 1.4.7.

Proof of Theorem 1.4.7. At first, note that because $R_{m} \leq R_{u n f}^{*}$, the guess $R_{u n f}^{*}$ will be among the ones we test. For the iteration of $R_{u n f}^{*}$, Lemmas 2.2 .11 and 2.2 .16 will clearly hold, thus ensuring that the returned solution has value $5 R_{u n f}^{*}$, and the constructed assignment satisfies all fairness constraints. The only thing left to analyze is the number of centers we end up using in the iteration of $R_{u n f}^{*}$. Note that by Observation 2.2.5 and the fact that the optimal unfair solution uses at most $k$ centers, we immediately get $\left|S_{I}\right|+\left|S_{N}\right| \leq k$. On the other hand, observe that the number of centers we use is in the worst case $2\left|S_{I}\right|+\left|S_{N}\right|$, and therefore at most $2 k$.

Finally, to conclude the proof, we just need to make sure that for a radius guess that resulted in $\left|S_{I}\right|+\left|S_{N}\right|>k$, we return an infeasibility message.

Although the result of Theorem 1.4.7 is interesting in the sense of showing a scenario with bounded PoF, it is not a true approximation algorithm, because we end up violating the number of chosen centers by a multiplicative factor of 2 . We now demonstrate another case, where we achieve a true feasible solution to EQCENTER-AG, that additionally enjoys a bounded PoF.

In this scenario, any two points are similar iff their distance is at most some value $R_{d}$, where $R_{d} \leq R_{u n f}^{*}$. Our algorithm here is actually identical to the one presented in the previous subsection, and the difficulty in proving Theorem 1.4.8 for it lies only on the analysis.

Proof of Theorem 1.4.8. At first, note that because $R_{d} \leq R_{u n f}^{*}$, the guess $R_{u n f}^{*}$ will be among the ones we test. For the iteration of $R_{u n f}^{*}$, Lemma 2.2.16 clearly holds. We will show that Lemma 2.2.11 will hold as well, and furthermore that Algorithm 5 will always pick just one center in each $G_{c}$ for $c \in S_{I}$. This will immediately imply that the returned solution has value at most $5 R_{u n f}^{*}$, all constraints (1.2) are satisfied, and the centers we end up using are exactly $\left|S_{I}\right|+\left|S_{N}\right|$. Finally, note that by Observation 2.2.5 and the fact that the optimal unfair solution uses at most $k$ centers, we will also have $\left|S_{I}\right|+\left|S_{N}\right| \leq k$.

Therefore, all we need to show is that for every $c \in S_{I}$, Algorithm 5 is able to find exactly one center that satisfies constraint (1.2) for all $j \in G_{c}$ (recall that $\mathcal{S}_{j} \subseteq G_{c}$ ). To do that, we prove that there exists an $x \in G_{c}$, such that that for all $j \in G_{c}$ we have $\sum_{j^{\prime} \in \mathcal{S}_{j}} d\left(j, j^{\prime}\right) \leq$ $\sum_{j^{\prime} \in \mathcal{S}_{j}} d\left(j^{\prime}, x\right)$. This suffices to prove the desired statement. To see why, assume that we make $x$ the chosen center of $G_{c}$, and assign all points of $G_{c}$ to $i$. Then for any point $j \in G_{c}$ and any $j^{\prime} \in \mathcal{S}_{j}$ we have $d(j, x) \leq d\left(j, j^{\prime}\right)+d\left(j^{\prime}, x\right)$ by the triangle inequality. Summing over all $j^{\prime} \in \mathcal{S}_{j}$
and using the defining property of $x$ gives:

$$
\begin{aligned}
d(j, x) & \leq \frac{1}{\left|\mathcal{S}_{j}\right|} \sum_{j^{\prime} \in \mathcal{S}_{j}} d\left(j, j^{\prime}\right)+\frac{1}{\left|\mathcal{S}_{j}\right|} \sum_{j^{\prime} \in \mathcal{S}_{j}} d\left(j^{\prime}, x\right) \\
& \leq \frac{2}{\left|\mathcal{S}_{j}\right|} \sum_{j^{\prime} \in \mathcal{S}_{j}} d\left(j^{\prime}, x\right) \\
& \leq \frac{\alpha}{\left|\mathcal{S}_{j}\right|} \sum_{j^{\prime} \in \mathcal{S}_{j}} d\left(j^{\prime}, x\right)
\end{aligned}
$$

The above inequality implies the satisfaction of (1.2) for point $j$.
For the sake of contradiction, assume now that for all $x \in G_{c}$ there exists a point $j \in G_{c}$ such that $\sum_{j^{\prime} \in \mathcal{S}_{j}} d\left(j, j^{\prime}\right)>\sum_{j^{\prime} \in \mathcal{S}_{j}} d\left(j^{\prime}, x\right)$. Based on this, we can create a dependency graph, where every point of $G_{c}$ is a vertex, and there is a directed edge from $x$ to $j$ if $\sum_{j^{\prime} \in \mathcal{S}_{j}} d\left(j, j^{\prime}\right)>$ $\sum_{j^{\prime} \in \mathcal{S}_{j}} d\left(j^{\prime}, x\right)$. The assumption for the contradiction implies that this dependency graph will contain a directed cycle $x_{1}, x_{2}, \ldots, x_{r}$, for which we have $\sum_{j^{\prime} \in \mathcal{S}_{x_{t}}} d\left(x_{t}, j^{\prime}\right)>\sum_{j^{\prime} \in \mathcal{S}_{x_{t}}} d\left(j^{\prime}, x_{t-1}\right)$ for all $t \in[2, r+1]$, assuming that $x_{r+1}=x_{1}$. If we add all the above inequalities we get

$$
\sum_{t=2}^{r+1} \sum_{j^{\prime} \in \mathcal{S}_{x_{t}}} d\left(j^{\prime}, x_{t}\right)>\sum_{t=2}^{r+1} \sum_{j^{\prime} \in \mathcal{S}_{x_{t}}} d\left(j^{\prime}, x_{t-1}\right)
$$

Now focus on any single $j^{\prime}$, and see that its contribution in the LHS of the above inequality is $A=\sum_{t: j^{\prime} \in \mathcal{S}_{x_{t}}} d\left(j^{\prime}, x_{t}\right)$, and in the RHS is $B=\sum_{t: j^{\prime} \in \mathcal{S}_{x_{t}}} d\left(j^{\prime}, x_{t-1}\right)$. We argue that $A>B$ is impossible, and thus reach a contradiction. If $A>B$, we can first subtract from both $A$ and $B$ the common terms appearing in the sums. Then, in what is left of $A$ we will only have terms $d\left(j^{\prime}, x_{t}\right)$ being added, for $j^{\prime} \in \mathcal{S}_{x_{t}}$. In what is left of $B$ we will only have terms $d\left(j^{\prime}, x_{t-1}\right)$ being added, but for which $j^{\prime} \notin \mathcal{S}_{x_{t-1}}$. Note also that the number of leftover terms is the same in both $A$ and
$B$. Since the similarity radius is the same for all points, for any two points $z, y \in G_{c}$, we have $d\left(j^{\prime}, z\right)<d\left(j^{\prime}, y\right)$ when $j^{\prime} \in \mathcal{S}_{z}$ and $j^{\prime} \notin \mathcal{S}_{y}$. Hence we reached the desired contradiction.

### 2.2.3 Solving the Assignment Problem

In this section we address the assignment problem for EQCENTER. Specifically, for an instance with $\alpha, k \geq 2$, if we are given the set of centers $S^{*}$ used in the optimal solution, can we efficiently find the optimal assignment $\phi^{*}: \mathcal{C} \mapsto S^{*}$ ? In other words, if $R^{*}$ is the value of the optimal solution, we want to compute $\phi^{*}$ such that 1) $\phi^{*}$ satisfies the appropriate fairness constraint for all clients, and 2) for every $j \in \mathcal{C}$ we have $d\left(j, \phi^{*}(j)\right) \leq R^{*}$. In what follows, we demonstrate in full detail a procedure that achieves this for EQCENTER-PP. A similar process can handle EQCENTER-AG, but for the sake of avoiding repeating the same arguments, we are only going to sketch the latter.

Before we proceed with our assignment algorithm for EQCENTER-PP, note that without loss of generality we can always assume that the optimal value $R^{*}$ is known. This is because there are only polynomially many options for it, and thus we can efficiently guess the optimal one. Our process is presented in Algorithm 7, and it works iteratively. The high-level idea is that it always maintains an assignment of value at most $R^{*}$, and in each iteration it corrects one violated fairness constraint. As we show later, a polynomial number of iterations suffices in order to reach a feasible assignment.

Lemma 2.2.17. Every time the condition of the while loop in Algorithm 7 is checked, we have $d(\phi(j), j) \geq d\left(\phi^{*}(j), j\right)$ for every $j \in \mathcal{C}$.

Proof. We are going to prove this via induction. For the first time we check the condition, the

```
Algorithm 7: Solving the assignment problem for EQCENTER-PP
    For every \(j \in \mathcal{C}\) set \(\phi(j) \leftarrow \arg \max _{i \in S^{*}: d(i, j) \leq R^{*}} d(i, j)\);
    while there exists a \(j \in \mathcal{C}\) with a \(j^{\prime} \in \mathcal{S}_{j}\) such that \(d(j, \phi(j))>\alpha d\left(j^{\prime}, \phi\left(j^{\prime}\right)\right.\) do
        Find such a pair \(j \in \mathcal{C}\) and \(j^{\prime} \in \mathcal{S}_{j}\);
        Let \(\Delta_{j, j^{\prime}}=\left\{i \in S^{*}: d(i, j)<d(j, \phi(j))\right.\) and \(\left.d(i, j) \leq \alpha d\left(j^{\prime}, \phi\left(j^{\prime}\right)\right)\right\}\);
        Set \(\phi(j) \leftarrow \arg \max _{i \in \Delta_{j, j^{\prime}}} d(i, j)\);
    end
    Return \(\phi\);
```

statement is obviously true by the way we initialized the mapping $\phi$ before the start of the loop, and the fact that $d\left(j, \phi^{*}(j)\right) \leq R^{*}$ for all $j \in \mathcal{C}$.

Consider now the $t^{\text {th }}$ time we check the condition, for which by the inductive hypothesis the statement of the lemma holds. If at that time no violated fairness constraint is found, then we are done. Hence, we need to focus on the case where the main body of the while loop is executed, and show that after the changes that occur in $\phi$, the statement will still be satisfied for the $(t+1)^{\text {th }}$ time we will check the condition.

Let $j_{t}$ be the client chosen at that iteration, with $j_{t}^{\prime} \in \mathcal{S}_{j_{t}}$ the client with $d\left(j_{t}, \phi\left(j_{t}\right)\right)>$ $\alpha d\left(j_{t}^{\prime}, \phi\left(j_{t}^{\prime}\right)\right)$. By the inductive hypothesis we have $d\left(j_{t}^{\prime}, \phi\left(j_{t}^{\prime}\right)\right) \geq d\left(j_{t}^{\prime}, \phi^{*}\left(j_{t}^{\prime}\right)\right)$. Combining the two previous inequalities gives $d\left(j_{t}, \phi\left(j_{t}\right)\right)>\alpha d\left(j_{t}^{\prime}, \phi^{*}\left(j_{t}^{\prime}\right)\right)$. Now because the optimal assignment satisfies $d\left(j_{t}, \phi^{*}\left(j_{t}\right)\right) \leq \alpha d\left(j_{t}^{\prime}, \phi^{*}\left(j_{t}^{\prime}\right)\right)$, we finally get $d\left(j_{t}, \phi\left(j_{t}\right)\right)>d\left(j_{t}, \phi^{*}\left(j_{t}\right)\right)$. In addition, we have $d\left(j_{t}, \phi^{*}\left(j_{t}\right)\right) \leq \alpha d\left(j_{t}^{\prime}, \phi^{*}\left(j_{t}^{\prime}\right)\right) \leq \alpha d\left(j_{t}^{\prime}, \phi\left(j_{t}^{\prime}\right)\right)$. Therefore, we see that $\phi^{*}\left(j_{t}\right) \in \Delta_{j_{t}, j_{t}}$. Let now $\phi^{\prime}\left(j_{t}\right)$ be the updated assignment for $j_{t}$ after the end of the iteration. From the way we update the assignment for $j_{t}$ and the fact that $\phi^{*}\left(j_{t}\right) \in \Delta_{j_{t}, j_{t}^{\prime}}$, we infer that $d\left(\phi^{\prime}\left(j_{t}\right), j_{t}\right) \geq d\left(\phi^{*}\left(j_{t}\right), j_{t}\right)$.

Theorem 2.2.18. Algorithm 7 terminates within $|\mathcal{C}| \cdot\left|S^{*}\right|$ iterations, with assignment $\phi$ satisfying:

1) $d(j, \phi(j)) \leq R^{*}$ for every $j \in \mathcal{C}$, and 2) $d(j, \phi(j)) \leq \alpha d\left(j^{\prime}, \phi\left(j^{\prime}\right)\right.$ for all $j \in \mathcal{C}$ and $j^{\prime} \in \mathcal{S}_{j}$.

Proof. From the condition of the while loop we know that when the algorithm terminates, the fairness constraints will be satisfied by the mapping $\phi$. Also, because we never assign a point to a center that is further than $R^{*}$ from it, we know that $\phi$ achieves the optimal value.

Now we are going to count the total possible number of iterations. We do that by considering how many times we changed the assignment of every single client $j$, i.e., how many times an iteration tried to fix one of $j$ 's violated constraints. By Lemma 2.2.17, we see that for any $j$ the minimum possible assignment distance we can provide to it is $d\left(j, \phi^{*}(j)\right)$. Observe that if at any moment $d(j, \phi(j))=d\left(j, \phi^{*}(j)\right)$, then Lemma 2.2.17 guarantees that $j$ 's assignment will never change again. This is because for every $j^{\prime} \in \mathcal{S}_{j}$ we always have $d\left(j^{\prime}, \phi\left(j^{\prime}\right)\right) \geq d\left(j^{\prime}, \phi^{*}\left(j^{\prime}\right)\right)$, and thus using the properties of the optimal assignment we get

$$
d(j, \phi(j))=d\left(j, \phi^{*}(j)\right) \leq \alpha d\left(j^{\prime}, \phi^{*}\left(j^{\prime}\right)\right) \leq \alpha d\left(j^{\prime}, \phi\left(j^{\prime}\right)\right)
$$

On the other hand, if at some point $d(j, \phi(j))>d\left(j, \phi^{*}(j)\right)$ then one of $j$ 's fairness constraints might be violated, and hence we might end up using an iteration to fix it. In this case, let $j^{\prime} \in \mathcal{S}_{j}$ the client causing the problematic situation. Note that Lemma 2.2.17 and the properties of the optimal solution ensure that $d\left(j_{t}, \phi^{*}\left(j_{t}\right)\right) \leq \alpha d\left(j_{t}^{\prime}, \phi^{*}\left(j_{t}^{\prime}\right)\right) \leq \alpha d\left(j_{t}^{\prime}, \phi\left(j_{t}^{\prime}\right)\right)$. Thus, for this iteration $\phi^{*}(j) \in \Delta_{j, j^{\prime}}$, and the new assignment distance of $j$ will be strictly smaller than the one it had at the beginning of the iteration. Thus, $j$ can be chosen in at most $\left|S^{*}\right|$ iterations.

The assignment procedure for EQCENTER-AG is almost identical to Algorithm 7, with the only difference being that we should look instead for violated constraints (1.2). In addition, the analysis of that algorithm remains identical to that of Algorithm 7.

### 2.2.4 Experimental Evaluation for EQCenter

We implemented all algorithms in Python 3.8 and ran our experiments on Intel Xeon (Ivy Bridge) E3-12 @ 2.4 GHz with 20 cores and 96 GB 1200 MHz DDR4 memory. Our code is publicly available: https://github.com/chakrabarti/equitable_clustering.

Datasets: We used 5 datasets from the UCI Machine Learning Repository [114], namely: (1) Bank-4,521 points [115], (2) Adult-32,561 points [116], (3) Creditcard-30,000 points [117], (4) Census1990-2,458,285 points [120] and (5) Diabetes-101,766 points [121]. From Adult, Creditcard, Census and Diabetes we uniformly subsampled 25, 000 points, and performed our experiments with respect to those sampled sets. In order to construct the distances between points, we removed non-numeric features, standardized each of the remaining features, took the Euclidean distances between these modified points, and then normalized the distances to be in $[0,1]$ for each dataset (by dividing the distances for a given dataset by the maximum distance between any two points).

Algorithms: We first implemented the two versions of the algorithm of Theorem 1.4.6, one solving EQCENTER-AG and the other EQCENTER-PP. We call Alg-AG the variant solving EQ-CEnter-AG, and Alg-PP the variant solving EQCEnter-PP. Furthermore, we implemented the algorithm of Theorem 1.4.7 and we refer to this as Pseudo-PoF-Alg. Finally, as baselines we used our own implementations of two "unfair" $k$-center algorithms, specifically the 2-approximation of Hochbaum and Shmoys [8] and the 2-approximation of Gonzalez [98].

Range of $k$ and value of fairness parameter $\alpha$ : We ran all of our experiments for every value of $k$ in $\{2,4,8,16,32,64,128\}$, and in all our simulations we set $\alpha=2$ for constraints (1.1) and (1.2). We did not test any other value for $\alpha$, since in practice $\alpha>2$ is unsuitable if
reasonably strong fairness considerations are at play.
Constructing the similarity sets: For each combination of dataset and value of $k$ that we are interested in, we need to construct the similarity sets $\mathcal{S}_{j}$, such that they $R_{m}=O\left(R_{u n f}^{*}\right)$ (refer to Section 1.4.2 for a discussion on why such a condition is realistic). Our first step in doing so, was utilizing the filtering procedure from [8], which for a given instance (combination of a dataset and a value $k$ ) returns a value $R_{f}$. If $R_{u n f}^{*}$ is the value of the optimal "unfair" $k$-center solution for the instance, the aforementioned filtering guarantees that $R_{f} \leq R_{u n f}^{*}$. Then, for each point $j$ we drew $R_{j}$ uniformly at random from $\left[0,2 R_{f}\right]$, and then set $\mathcal{S}_{j}=\left\{j^{\prime} \mid d\left(j, j^{\prime}\right) \leq R_{j}\right\}$. There were two reasons for constructing the sets $\mathcal{S}_{j}$ in this way. At first, this approach guarantees the realistic $R_{m}=O\left(R_{u n f}^{*}\right)$ condition. Second, this approach forces non-uniformity in the values of $R_{j}$, and thus we are able to test our algorithms in the most general setting (for instance the uniform setting described in Theorem 1.4.8 is more restricted and less realistic).

Evaluated Metrics: Let $S$ be the set of chosen centers and $\phi: \mathcal{C} \mapsto S$ the corresponding assignment function, that constituted the solution we got when we ran some particular algorithm on some problem instance. The quantities we evaluate are:

- Maximum assignment distance $\left(\max _{j \in \mathcal{C}} d(j, \phi(j))\right.$ : This is the actual objective function value of the returned solution.
- Satisfaction of constraint (1.1): Here for each $j$ we define $f_{j}^{\mathrm{PP}}=\max _{j^{\prime} \in \mathcal{S}_{j}} \frac{d(j, \phi(j))}{d\left(j^{\prime}, \phi\left(j^{\prime}\right)\right)}$.
- Satisfaction of constraint (1.2): Here for each $j$ we define $f_{j}^{\mathrm{AG}}=\frac{\left|\mathcal{S}_{j}\right| d(j, \phi(j))}{\sum_{j^{\prime} \in \mathcal{S}_{j}} d\left(j^{\prime}, \phi\left(j^{\prime}\right)\right)}$.

We now present our results that involve running all 5 mentioned algorithms on the Adult dataset. The corresponding plots for the other four datasets can be found in Appendix A, and they exhibit the exact behavior as the ones displayed here. Further, the maximum runtime encountered


Figure 2.3: Adult: Maximum assignment distance for all algorithms
in all our simulations was approximately 30 minutes (running Alg-PP on Census1990), and the bottleneck in all executions was computing the pairwise distances and not running the algorithms.

In Figure 2.3 we present the maximum assignment distance as a function of $k$ for all algorithms. At first, we observe that even our algorithms with no PoF guarantees, i.e., Alg-PP and Alg-AG, perform very well in terms of an empirical PoF with respect to the baseline solutions. In addition, we want to compare the objective values of Alg-PP and Alg-AG. Recall that since a solution to EQCENTER-PP also constitutes a solution to EQCENTER-AG, we are theoretically expecting Alg-AG to perform better. However, we see that in practice there is no clear-cut winner, and hence the use of Alg-PP is highly recommended, since the notion of fairness guaranteed by that algorithm is much stronger.

In Figure 2.4 we demonstrate how all algorithms perform in terms of the fairness con-


Figure 2.4: Adult: Satisfaction of fairness constraints
straints. ${ }^{1}$ Figure 2.4a shows $\max _{j} f_{j}^{P P}$ as a function of $k$ for our two algorithms for EQCENTERPP, i.e., Alg-PP and Pseudo-PoF-Alg. Here we see that as the theory suggests, our algorithms always satisfy constraint (1.1) and have $\max _{j} f_{j}^{P P} \leq 2$. On the other hand, Figure 2.4 b shows $\max _{j} f_{j}^{P P}$ as a function of $k$ for the baselines. Here we see that the baselines are far from satisfying constraint (1.1), and specifically that there exist points that are treated very unfairly. Finally, Figure 2.4 c shows $\max _{j} f_{j}^{A G}$ as a function of $k$ for all algorithms that can be potentially used for EQCENTER-AG. Here we see that our algorithms again satisfy the corresponding constraint (1.2), and furthermore have a better $\max _{j} f_{j}^{A G}$ value compared to the baselines. Finally, in the AG case the baselines seem to perform much better compared to the PP case, and this is reasonable because the notion of fairness described by (1.2) is much weaker. Nonetheless, in most cases the baselines are not able to satisfy (1.2).

In Figures 2.5a and 2.5b we are interested in the percentage of points for which baselines do not satisfy the appropriate fairness constraint. Specifically, Figure 2.5 a demonstrates that for the stronger notion of PP-fairness, a substantial percentage of points gets unfair treatment $\left(f_{j}^{P P}>2\right)$.

[^4]

Figure 2.5: Adult: Amount of constraint violation

On the other hand, for the weaker (1.2), the baselines do much better. Nonetheless, even if one is interested only in the weaker AG concept of fairness, they should not use the baselines. Even one unfairly treated point goes against the very nature of individual fairness.

Finally, in Figure 2.5 c we see by how much Pseudo-PoF-Alg violates the constraint $|S| \leq k$ on the set of chosen centers (recall that in theory Pseudo-PoF-Alg yields $|S| \leq 2 k$ ). Here we plot the ratio of the number of centers used by the algorithm over the given value $k$, and see that in practice Pseudo-PoF-Alg does not actually incur any violation.

## Chapter 3: Addressing the Fair Cuts Problems

### 3.1 Reduction to Tree Instances

Here we show how both DemFairCut and IndFairCut can be effectively reduced to solving an appropriate problem on a tree instance. To do this, we use the following lemma.

Lemma 3.1.1. [54] For any undirected graph $G=(V, E)$ with edge costs $w_{e} \geq 0$, we can efficiently construct a collection of trees $T_{1}=\left(V, E_{1}\right), T_{2}=\left(V, E_{2}\right), \ldots, T_{k}=\left(V, E_{k}\right)$ with tree $T_{i}$ having an edge-cost function $w^{i}: E_{i} \mapsto \mathbb{R}_{\geq 0}$, and find non-negative multipliers $\left(\lambda_{1}, \ldots, \lambda_{k}\right)$, such that $\sum_{i=1}^{k} \lambda_{i}=1$ and $k=\operatorname{poly}(|V|)$. Moreover, for any $S \subseteq V$ let $\delta(S)$ denote the set of edges in $E$ with exactly one endpoint in $S$, and $\delta_{i}(S)$ denote the set of edges in $E_{i}$ with exactly one endpoint in $S$. Then, for every $S \subseteq V$ we have:

1. $w(\delta(S)) \leq w^{i}\left(\delta_{i}(S)\right)$ for every $i \in[k]$
2. $\sum_{i=1}^{k} \lambda_{i} w^{i}\left(\delta_{i}(S)\right) \leq O(\log n) w(\delta(S))$

Definition 3.1.2. We call an algorithm for DEmFairCut ( $\rho, \alpha$ )-bicriteria, if for any given problem instance $\mathcal{I}=\left\{V, E, s, w, V_{1}, \ldots, V_{\gamma}, \vec{f}\right\}$ with optimal value $O P T_{\mathcal{I}}$, it returns a solution $F$ such that 1) $w(F) \leq \rho O P T_{\mathcal{I}}$, and 2) $\left|\operatorname{prot}(V, E \backslash F, s) \cap V_{h}\right| \geq \alpha f_{h} n_{h}, \forall h \in[\gamma]$.

Lemma 3.1.3. If we have a $(\rho, \alpha)$-bicriteria algorithm for DEMFAIRCUT in trees, we can get a $(\rho \cdot O(\log n), \alpha)$-bicriteria algorithm for DEMFAIRCUT in general graphs.

Proof. If $\mathcal{I}=\left\{V, E, s, w, V_{1}, \ldots, V_{\gamma}, \vec{f}\right\}$ is the general instance, we first apply the result of Lemma 3.1.1 in order to get a collection of trees $T_{1}=\left(V, E_{1}\right), \ldots, T_{k}=\left(V, E_{k}\right)$, where each tree $T_{i}$ has an associated edge weight function $w^{i}$. We then use the given algorithm and solve DEMFAIRCUT in each tree instance $\mathcal{I}_{i}=\left\{V, E_{i}, s, w^{i}, V_{1}, \ldots, V_{\gamma}, \vec{f}\right\}$, and get a solution $F_{i} \subseteq E_{i}$ in return. For the solution $F_{i}$ we compute for $\mathcal{I}_{i}$, let $X_{i}=\operatorname{prot}\left(V, E_{i} \backslash F_{i}, s\right)$, and note that the properties of the algorithm ensure $\left|X_{i} \cap V_{h}\right| \geq \alpha f_{h} n_{h}, \forall h \in[\gamma]$.

After running the algorithm in each tree, let $T_{m}$ be the tree with $m=\arg \min _{i} w\left(\delta\left(X_{i}\right)\right)$. We then set our solution for the general graph to be $\delta\left(X_{m}\right)$. This means that in our general solution $X_{m} \subseteq \operatorname{prot}\left(V, E \backslash \delta\left(X_{m}\right), s\right)$. Combining this observation with the fact that $\left|X_{m} \cap V_{h}\right| \geq$ $\alpha f_{h} n_{h}$ for all $h \in[\gamma]$, implies that in the general graph solution we again satisfy all demographic constraints up to an $\alpha$ fraction. We now only have to reason about the cost of $\delta\left(X_{m}\right)$.

Let $X^{*}$ be the set of vertices not connected to $s$ in the optimal solution of $\mathcal{I}$. If $O P T$ is the value of the latter, then $w\left(\delta\left(X^{*}\right)\right) \leq O P T$. Also, since $X^{*}$ satisfies all $\gamma$ demographic constraints exactly, the set $\delta_{i}\left(X^{*}\right)$ is a feasible solution for $\mathcal{I}_{i}$, and $O P T_{\mathcal{I}_{i}} \leq w^{i}\left(\delta_{i}\left(X^{*}\right)\right)$. Since $\delta_{i}\left(X_{i}\right) \subseteq F_{i}$ :

$$
\begin{align*}
w^{i}\left(\delta_{i}\left(X_{i}\right)\right) & \leq \rho \cdot O P T_{\mathcal{I}_{i}} \\
& \leq \rho \cdot w^{i}\left(\delta_{i}\left(X^{*}\right)\right) \tag{3.1}
\end{align*}
$$

Using the definition of $m$ and the first property of the trees from Lemma 3.1.1 gives

$$
\begin{align*}
w\left(\delta\left(X_{m}\right)\right) & \leq \sum_{i=1}^{k} \lambda_{i} w\left(\delta\left(X_{i}\right)\right) \\
& \leq \sum_{i=1}^{k} \lambda_{i} w^{i}\left(\delta_{i}\left(X_{i}\right)\right) \tag{3.2}
\end{align*}
$$

Combining (3.1), (3.2) and the second property of Lemma 3.1.1 yields

$$
\begin{aligned}
w\left(\delta\left(X_{m}\right)\right) & \leq \rho \sum_{i=1}^{k} \lambda_{i} w^{i}\left(\delta_{i}\left(X^{*}\right)\right) \\
& \leq \rho O(\log n) w\left(\delta\left(X^{*}\right)\right) \leq \rho O(\log n) O P T
\end{aligned}
$$

Our approach for tackling INDFAIRCUT uses as a black-box an algorithm for a new problem, which we call AUXCUT and we formally define below. In order to get an algorithm for general instances of AUXCUT, we again use a reduction to trees.

AuxCut: We are given an undirected graph $G=(V, E)$, a designated vertex $s \in V$, a budget $B>0$, and a target value $T \in \mathbb{N}_{\geq 0}$. In addition, each $e \in E$ has a weight $w_{e} \geq 0$, and each vertex $v \in V \backslash\{s\}$ has a value $a_{v} \geq 0$. The goal is to find a cut $F$ with $w(F) \leq B$ and $|\operatorname{prot}(V, E \backslash F, s)| \geq T$, that maximizes $a(\operatorname{prot}(V, E \backslash F, s))$.

Definition 3.1.4. We say that an algorithm is $(1,1, \rho)$-bicriteria for AUXCUT, if for any given instance $\mathcal{I}=(V, E, B, T, w, s, a)$ of the problem with optimal value $O P T_{\mathcal{I}}$, it returns $F \subset E$, such that 1) $w(F) \leq \rho B$, 2) $|\operatorname{prot}(V, E \backslash F, s)| \geq T$ and 3) $a(\operatorname{prot}(V, E \backslash F, s)) \geq O P T_{\mathcal{I}}$.

Lemma 3.1.5. If we have a $(1,1, \rho)$-bicriteria algorithm for AUXCUT in tree instances, we can devise a $(1,1, \rho \cdot O(\log n))$-bicriteria algorithm for AUXCUT in general graphs.

Proof. Let $\mathcal{I}=(V, E, B, T, w, s, a)$ be an instance of AuxCut for a general graph. We first apply the result of Lemma 3.1.1 in order to get a collection of trees $T_{i}=\left(V, E_{i}\right)$ with edge-weight functions $w^{i}$. Then, for each such tree we create an instance $\mathcal{I}_{i}=\left(V, E_{i}, B \cdot O(\log n), T, w^{i}, s, a\right)$, and we use the given bicriteria algorithm to solve AuxCut on it. Let $F_{i} \subseteq E_{i}$ the solution we get for $\mathcal{I}_{i}$, and for notational convenience let again $X_{i}=\operatorname{prot}\left(V, E_{i} \backslash F_{i}, s\right)$. After that, we find
the tree $T_{m}$ with $m=\arg \max _{i} a\left(X_{i}\right)$, and we set our solution for the general graph to be $\delta\left(X_{m}\right)$. This means that in our general solution we again get $X_{m} \subseteq \operatorname{prot}\left(V, E \backslash \delta\left(X_{m}\right), s\right)$.

At first, because of the properties of the algorithm used on $\mathcal{I}_{i}$, we have $\left|X_{m}\right| \geq T$, and therefore even in our solution for the general graph we end up saving at least $T$ vertices.

Furthermore, because $\delta_{i}\left(X_{i}\right) \subseteq F_{i}$, the properties of the algorithm give $w^{i}\left(\delta_{i}\left(X_{i}\right)\right) \leq$ $\rho \cdot O(\log n) \cdot B$ for every $i$. From the first property in Lemma 3.1.1 we thus get

$$
\begin{aligned}
w\left(\delta\left(X_{m}\right)\right) & \leq w^{m}\left(\delta_{m}\left(X_{m}\right)\right) \\
& \leq \rho \cdot O(\log n) \cdot B
\end{aligned}
$$

To conclude we need to show that $a\left(X_{m}\right) \geq O P T_{\mathcal{I}}$, where $O P T_{\mathcal{I}}$ the value of the optimal solution of $\mathcal{I}$. Let also $X^{*}$ be the set of vertices disconnected from $s$ in the optimal solution of $\mathcal{I}$. Since $X^{*}$ is the optimal such set of vertices, we have $\left|X^{*}\right| \geq T$ and $w\left(\delta\left(X^{*}\right)\right) \leq B$. Let $m^{*}=\arg \min _{i} w^{i}\left(\delta_{i}\left(X^{*}\right)\right)$. The definition of $m^{*}$ and the second property of Lemma 3.1.1 give

$$
\begin{aligned}
w^{m^{*}}\left(\delta_{m^{*}}\left(X^{*}\right)\right) & \leq \sum_{i=1}^{k} \lambda_{i} w^{i}\left(\delta_{i}\left(X^{*}\right)\right) \\
& \leq O(\log n) w\left(\delta\left(X^{*}\right)\right) \\
& \leq B \cdot O(\log n)
\end{aligned}
$$

Hence $\delta_{m^{*}}\left(X^{*}\right)$ is feasible for $\mathcal{I}_{m^{*}}$ (recall that $\left|X^{*}\right| \geq T$ ), and since the given algorithm is a $(1,1, \rho)$-bicriteria we get

$$
a\left(X_{m}\right) \geq a\left(X_{m^{*}}\right) \geq O P T_{\mathcal{I}_{m^{*}}} \geq a\left(X^{*}\right)=O P T_{\mathcal{I}}
$$

### 3.2 Solving DemFairCut

In this section we tackle DEMFAIRCUT and present two algorithms for it. The first works for a constant $\gamma$, and is an $O(\log n)$-approximation. The second addresses the case of an arbitrary $\gamma$, and for any $\epsilon>0$ it is an $\left(O\left(\frac{\log n \log \gamma}{\epsilon^{2} \cdot \min _{h} f_{h}}\right), 1-\epsilon\right)$-bicriteria one.

### 3.2.1 Solving DemFairCut for $\gamma=O(1)$

Given Lemma 3.1.3, we can focus on only solving the problem in tree instances. Specifically, we show that when $\gamma=O(1)$ the problem in trees can be solved optimally via dynamic programming. Without loss of generality, we can also assume that the given tree is rooted at $s$ and it is binary. For details on why this assumption is safe to use, we refer the reader to Lemma 15.18 from [122]. Before we describe our approach we need some additional notation. For a vertex $v$, let $\phi_{h}(v)=1$ if $v \in V_{h}$ and 0 otherwise.

Our dynamic programming algorithm is based on a table $M$, where $M\left[v, k_{1}, k_{2}, \ldots, k_{\gamma}\right]$ represents the minimum cost of a cut in the subtree rooted at $v$, so that there are exactly $k_{h}$ nodes from $V_{h}$ that are connected to $v$. Let $v_{r}$ be the right child of $v$, and let $v_{\ell}$ be the left child of $v$. Observe that the optimal either cuts neither of the edges from $v$ to its children, just the left edge, just the right edge, or both. So, we set $M\left[v, k_{1}, k_{2}, \ldots, k_{\gamma}\right]$ to the minimum of the following:

1. $\min \left\{M\left[v_{\ell}, k_{1}^{\ell}, k_{2}^{\ell}, \ldots, k_{\gamma}^{\ell}\right]+M\left[u_{r}, k_{1}^{r}, k_{2}^{r}, \ldots, k_{\gamma}^{r}\right]: k_{h}^{\ell}+k_{h}^{r}+\phi_{h}(v)=k_{h} \forall h \in[\gamma]\right\}$
2. $\min \left\{w_{\left(v, v_{\ell}\right)}+M\left[v_{r}, k_{1}^{\prime}, k_{2}^{\prime}, \ldots, k_{\gamma}^{\prime}\right]: k_{h}^{\prime}+\phi_{h}(v)=k_{h} \forall h \in[\gamma]\right\}$
3. $\min \left\{w_{\left(v, v_{r}\right)}+M\left[v_{\ell}, k_{1}^{\prime}, k_{2}^{\prime}, \ldots, k_{\gamma}^{\prime}\right]: k_{h}^{\prime}+\phi_{h}(v)=k_{h} \forall h \in[\gamma]\right\}$
4. $w_{\left(v, v_{r}\right)}+w_{\left(v, v_{\ell}\right)}$ if $k_{h}=\phi_{h}(v)$ for all $h \in[\gamma],+\infty$ otherwise.

The first case above corresponds to cutting neither of the edges $\left(v, v_{r}\right),\left(v, v_{\ell}\right)$, the second to cutting only $\left(v, v_{\ell}\right)$, the third to cutting only $\left(v, v_{r}\right)$, and the fourth to cutting both.

To fill in $M$, we begin by initializing $M\left[v, \phi_{1}(v), \phi_{2}(v), \ldots, \phi_{\gamma}(v)\right]=0$ for all leaves $v$ of the tree, and set all other table entries to $+\infty$. Then we proceed by filling the table bottom-up. There are at most $O\left(n^{\gamma+1}\right)$ table entries, and to compute each one we need to access at most $2 n^{\gamma}$ other ones. Thus, the total runtime is $O\left(n^{2 \gamma+1}\right)$. Finally, in order to find the optimal cut, we look for the minimum entry $M\left[s, k_{1}, \ldots, k_{\gamma}\right]$, such that $k_{h} \leq\left(1-f_{h}\right) n_{h}$ for all $h \in[\gamma]$.

Theorem 3.2.1. When $\gamma$ is a constant, we have an optimal dynamic programming algorithm for DEMFAIRCUT in trees, running in time $O\left(n^{2 \gamma+1}\right)$.

Combining Theorem 3.2.1 with Lemma 3.1.3, our approach achieves the following.

Theorem 3.2.2. For $\gamma=O(1)$, we give a $O(\log n)$-approximation algorithm for DEMFAIRCUT.

### 3.2.2 Solving DemFaircut for an Arbitrary $\gamma$

Given Lemma 3.1.3, we again focus on instances $\mathcal{I}=\left\{V, E, s, w, V_{1}, \ldots, V_{\gamma}, \vec{f}\right\}$, where the underlying graph $T=(V, E)$ is a tree. Moreover, we can assume without loss of generality that the tree is rooted at $s$. Before we proceed with the description of our algorithm, we need some more notation. For every $v \in V$ let $P(s, v) \subseteq E$ be the unique path from $s$ to $v$ in the tree, and $\ell(v)=|P(s, v)|$. In addition, for every $e=(u, v) \in E$ let $P_{e}=P(s, r(e))$, with $r(e)=\arg \min _{z \in\{u, v\}} \ell(z)$. In words, $P_{e}$ contains the edges of the path that starts from $s$ and
finishes just before reaching $e$. The following LP is then a valid relaxation of our problem.

$$
\begin{array}{lr}
\min \sum_{e \in E} w_{e} \cdot x_{e} & \\
y_{v}=\sum_{e \in P(s, v)} x_{e} & \forall v \in V \\
\sum_{v \in V_{h}} y_{v} \geq f_{h} \cdot n_{h} & \forall h \in[\gamma] \\
0 \leq y_{v}, x_{e} \leq 1 & \forall v \in V, e \in E \tag{3.6}
\end{array}
$$

In the integral version of LP (3.3)-(3.6), $x_{e}=1$ iff edge $e$ is included in the cut. Now notice that because the underlying graph is a tree and the edge weights are non-negative, for any $v \in V$ the optimal solution would not choose more than one edge from $P(s, v)$. Therefore, by constraints (3.4) and (3.6) we see that $y_{v}=1$ iff $v$ is separated from $s$ in the optimal outcome. Consequently, constraint (3.5) naturally captures the demographic covering requirements.

Our approach begins by solving LP (3.3)-(3.6) in order to get a fractional solution $x, y$. We then apply the following dependent randomized rounding scheme. We consider the edges of the tree in non-decreasing order of $\left|P_{e}\right|$, and for an edge $e$ for which no other edge in $P_{e}$ is already chosen for the cut, we remove it with probability $x_{e} /\left(1-x\left(P_{e}\right)\right)$ if $x\left(P_{e}\right)<1$. The latter action makes sense because for every $e^{\prime} \in P_{e}$ we have $\left|P_{e^{\prime}}\right|<\left|P_{e}\right|$, and hence $e^{\prime}$ is considered before $e$ in the given ordering. Further, if an edge $e$ is chosen to be placed in the cut, then all $v \in V$ with $e \in P(s, v)$ are now disconnected from $s$. In addition, observe that due to the dependent nature of this process, no path $P(s, v)$ will have more than one edge of it in the solution.

Algorithm 8 demonstrates all necessary details of the rounding, with $X_{e}$ being an indicator random variable denoting whether or not $e$ is included in the solution, and $Y_{v}$ an indicator random

```
Algorithm 8: Randomized Rounding for LP (3.3)-(3.6)
    For every \(e \in E\) set \(X_{e} \leftarrow 0\), and for all \(v \in V\) set \(Y_{v} \leftarrow 0\);
    for all \(e \in E\) in non-decreasing order of \(\left|P_{e}\right|\) do
        if \(x\left(P_{e}\right)<1\) and \(X_{e^{\prime}}=0\) for all \(e^{\prime} \in P_{e}\) then
            Set \(X_{e} \leftarrow 1\) with probability \(x_{e} /\left(1-x\left(P_{e}\right)\right)\);
            if \(X_{e}=1\) then
                Set \(Y_{v} \leftarrow 1\) for all \(\{v \in V: e \in P(s, v)\} ;\)
            end
        end
    end
```

variable that is 1 iff $v$ is disconnected from $s$ in the final outcome.

Lemma 3.2.3. When we decide to include $e \in E$ in the cut, we do so with a valid probability.

Proof. Let $e=(u, v)$, and without loss of generality assume $l(u)<l(v)$. This means that $P_{e}=P(s, u)$ and $P(s, v)=P(s, u) \cup\{e\}$. In addition, to consider a randomized decision for $e$ we should also have $x\left(P_{e}\right)<1$. Using constraints (3.4) and (3.6) for $v$ we get:

$$
x_{e}+\sum_{e^{\prime} \in P_{e}} x_{e^{\prime}} \leq 1 \Longrightarrow \frac{x_{e}}{1-x\left(P_{e}\right)} \leq 1
$$

Lemma 3.2.4. For every $e \in E$ and $v \in V$, we have $\operatorname{Pr}\left[X_{e}=1\right]=x_{e}$ and $\operatorname{Pr}\left[Y_{v}=1\right]=y_{v}$.

Proof. Let us begin with an $e \in E$ for which we never made a random decision because $x\left(P_{e}\right) \geq$ 1, and hence $X_{e}=0$. If $e=(u, v)$ with $l(u)<l(v)$, then $P_{e}=P(s, u)$ and $P(s, v)=P(s, u) \cup$ $\{e\}$. Because of constraints (3.4) and (3.6) for $u$ we first get $x\left(P_{e}\right)=1$. Therefore, constraints (3.4) and (3.6) applied this time for $v$ yield $x_{e}=0$, which indeed gives $\operatorname{Pr}\left[X_{e}=1\right]=x_{e}$.

Now let us consider an edge $e$ with $x\left(P_{e}\right)<1$. Because for each $e^{\prime} \in P_{e}$ we have $P_{e^{\prime}} \subset P_{e}$, we also get $x\left(P_{e^{\prime}}\right)<1$. The latter means that for all other edges in $P_{e}$ a random decision potentially takes place. Furthermore, analysis of the algorithm's actions shows that $\operatorname{Pr}\left[X_{e}=1\right]$
is equal to

$$
\begin{align*}
& \operatorname{Pr}\left[X_{e}=1 \mid X_{e^{\prime}}=0 \forall e^{\prime} \in P_{e}\right] \cdot \operatorname{Pr}\left[X_{e^{\prime}}=0 \forall e^{\prime} \in P_{e}\right] \\
& =\frac{x_{e}}{1-\sum_{e^{\prime} \in P_{e}} x_{e^{\prime}}} \prod_{e^{\prime} \in P_{e}}\left(1-\frac{x_{e^{\prime}}}{1-\sum_{e^{\prime \prime} \in P_{e^{\prime}}} x_{e^{\prime \prime}}}\right) \tag{3.7}
\end{align*}
$$

Let $e_{1}, \ldots, e_{m}$ the edges of $P_{e}$ in increasing order of $\left|P_{e_{j}}\right|$. Then because $P_{e_{j}}=\left\{e_{j^{\prime}} \mid j^{\prime}<j\right\}$, expression (3.7) can be rewritten as a telescopic product of fractions:

$$
\frac{x_{e}}{1-\sum_{j=1}^{m} x_{e_{j}}} \prod_{j=1}^{m}\left(1-\frac{x_{e_{j}}}{1-\sum_{i=1}^{j-1} x_{e_{i}}}\right)=x_{e}
$$

As for a vertex $v \in V$, we have $\operatorname{Pr}\left[Y_{v}=1\right]=\operatorname{Pr}\left[\exists e \in P(s, v): X_{e}=1\right]$ because there is a unique path from $s$ to it. Moreover, since our rounding will never put more than one edges of $P(s, v)$ in the cut, for all $S \subseteq P(s, v)$ with $|S| \geq 2$ we get $\operatorname{Pr}\left[X_{e}=1, \forall e \in S\right]=0$. Hence, by the inclusion-exclusion principle $\operatorname{Pr}\left[\exists e \in P(s, v): X_{e}=1\right]=\sum_{e \in P(s, v)} \operatorname{Pr}\left[X_{e}=1\right]=$ $\sum_{e \in P(s, v)} x_{e}=y_{v}$, where the last equality follows from constraint (3.4).

We will now analyze the satisfaction of the coverage constraints for the different demographics. If $S_{h}$ is the number of vertices from $V_{h}$ that are not connected to $s$ in the solution, we see that $S_{h}=\sum_{v \in V_{h}} Y_{v}$. Using Lemma 3.2.4 and constraint (3.5) gives $\mathbb{E}\left[S_{h}\right] \geq f_{h} n_{h}$. We thus need to calculate how much can $S_{h}$ deviate from $\mathbb{E}\left[S_{h}\right]$. For that we need the next two lemmas.

Lemma 3.2.5. [123] Let $Z_{1}, \ldots, Z_{m}$ be Bernoulli random variables, where $\operatorname{Pr}\left[Z_{i}=1\right]=z_{i}$ for all $i \in[m]$. Let $\Gamma$ the dependency graph on the $Z_{i}$. For $i \neq j, Z_{i}$ and $Z_{j}$ are dependent if there exists an edge between them in $\Gamma$, and we denote that as $i \sim j$. Let also $Z=\sum_{i=1}^{m} Z_{i}, \mu=\mathbb{E}[Z]$,
$\Delta=\sum_{\{i, j\}: i \sim j} \operatorname{Pr}\left[Z_{i}=Z_{j}=1\right], \delta_{i}=\sum_{j \sim i} z_{j}$ and $\delta=\max _{i} \delta_{i}$. Then for any $\epsilon \in[0,1]$

$$
\operatorname{Pr}[Z \leq(1-\epsilon) \mu] \leq \exp \left(-\min \left(\frac{\epsilon^{2} \cdot \mu^{2}}{8 \Delta+2 \mu}, \frac{\epsilon \cdot \mu}{6 \delta}\right)\right)
$$

Lemma 3.2.6. For all $m \in \mathbb{N}_{>0}$ and any sequence of non-negative numbers $a_{1}, a_{2}, \ldots$ we have

$$
\sum_{i=1}^{m-1}(m-i) a_{i} \leq m \sum_{i=1}^{m} a_{i}
$$

Proof. We prove the statement via induction on $m$. For $m=1$ it is trivial. Suppose that the lemma holds up to some $m=k$. We then prove it for $m=k+1$ :

$$
\begin{aligned}
\sum_{i=1}^{k+1-1}(k+1-i) a_{i} & =\sum_{i=1}^{k}\left((k-i) a_{i}+a_{i}\right) \\
& =\sum_{i=1}^{k}(k-i) a_{i}+\sum_{i=1}^{k} a_{i} \\
& =\sum_{i=1}^{k-1}(k-i) a_{i}+\sum_{i=1}^{k} a_{i} \\
& \leq k \sum_{i=1}^{k} a_{i}+\sum_{i=1}^{k} a_{i} \\
& \leq(k+1) \sum_{i=1}^{k} a_{i} \\
& \leq(k+1) \sum_{i=1}^{k+1} a_{i}
\end{aligned}
$$

The first inequality uses the inductive hypothesis, while the last the fact that $a_{k+1} \geq 0$.
Lemma 3.2.7. For all $h \in[\gamma]$ and any $\epsilon \in[0,1]$, we have $\operatorname{Pr}\left[S_{h} \leq(1-\epsilon) \mathbb{E}\left[S_{h}\right]\right] \leq e^{\frac{-\epsilon^{2} \cdot f_{h}}{10}}$.

Proof. Due to Lemma 3.2.4, the variables $Y_{v}$ for $v \in V_{h}$ are Bernoulli with $\operatorname{Pr}\left[Y_{v}=1\right]=y_{v}$.

Because of the tree structure they are also to some extent dependent. Our goal here is to apply Lemma 3.2.5 for $S_{h}$, and towards that end we need to upper bound the dependency factors $\delta, \Delta$. Since we do not know exactly the underlying dependency graph $\Gamma$, in what follows we assume that all pairs $Y_{v}, Y_{v^{\prime}}$ are dependent. We begin by bounding the parameter $\Delta$ of Lemma 3.2.5.

$$
\begin{aligned}
\Delta & \leq \sum_{\left\{v, v^{\prime}\right\} \in V_{h}} \operatorname{Pr}\left[Y_{v}=Y_{v^{\prime}}=1\right] \\
& \leq \sum_{\left\{v, v^{\prime}\right\} \in V_{h}} \min \left(\operatorname{Pr}\left[Y_{v}=1\right], \operatorname{Pr}\left[Y_{v^{\prime}}=1\right]\right) \\
& =\sum_{\left\{v, v^{\prime}\right\} \in V_{h}} \min \left(y_{v}, y_{v^{\prime}}\right)
\end{aligned}
$$

Now let $a_{1}, a_{2}, \ldots, a_{n_{h}}$ be the values $y_{v}$ for all $v \in V_{h}$ in non-decreasing order. Then we have:

$$
\sum_{\left\{v, v^{\prime}\right\} \in V_{h}} \min \left(y_{v}, y_{v^{\prime}}\right)=\sum_{i=1}^{n_{h}-1}\left(n_{h}-i\right) a_{i} \leq n_{h} \sum_{i=1}^{n_{h}} a_{i}=n_{h} \cdot \mathbb{E}\left[S_{h}\right]
$$

To get the first inequality we used Lemma 3.2.6. Therefore, we get $\Delta \leq n_{h} \cdot \mathbb{E}\left[S_{h}\right]$. Moreover, a straightforward upper bound for each $\delta_{v}$ is $\delta_{v} \leq \sum_{u \in V_{h}} y_{u}=\mathbb{E}\left[S_{h}\right]$. Thus, $\delta \leq \mathbb{E}\left[S_{h}\right]$. Finally, we also need bounds for the following two quantities, where $\mu=\mathbb{E}\left[S_{h}\right]$ :

$$
\begin{gathered}
\frac{\epsilon^{2} \cdot \mu^{2}}{8 \Delta+2 \mu} \geq \frac{\epsilon^{2} \cdot \mu^{2}}{8 \mu \cdot n_{h}+2 \mu}=\frac{\epsilon^{2} \cdot \mu}{8 n_{h}+2} \geq \frac{\epsilon^{2} \cdot n_{h} \cdot f_{h}}{8 n_{h}+2} \geq \frac{\epsilon^{2} \cdot f_{h}}{10} \\
\frac{\epsilon \cdot \mu}{6 \delta} \geq \frac{\epsilon \cdot \mu}{6 \mu}=\frac{\epsilon}{6}
\end{gathered}
$$

Since $\frac{\epsilon}{6} \geq \frac{\epsilon^{2} \cdot f_{h}}{10}$ for any $\epsilon, f_{h} \in[0,1]$, Lemma 3.2.5 gives the desired bound.

To conclude, suppose that for some $\beta \geq 2$, we repeat Algorithm 8 independently $N=$ $\frac{10 \log \gamma^{\beta}}{\epsilon^{2} \cdot \min _{h} f_{h}}$ times, and in each run $t$ of it (with $t \in[N]$ ) we compute a set of edges $F_{t}$ that are chosen to be removed. Our final solution is set to be $F=\bigcup_{t} F_{t}$. Then we have the following.

Theorem 3.2.8. For DemFaircut in trees and any $\epsilon \in(0,1)$, we give a $\left(O\left(\frac{\log \gamma}{\epsilon^{2} \min _{h} f_{h}}\right), 1-\epsilon\right)$ bicriteria algorithm that runs in expected polynomial time.

Proof. Focus on a specific $h$, and let $S_{h}^{t}$ the random variable denoting the number of nodes of $V_{h}$ separated from $s$ in $\left(V, E \backslash F_{t}\right)$. By Lemma 3.2.7 and the independent nature of the runs:

$$
\operatorname{Pr}\left[S_{h}^{t} \leq(1-\epsilon) \mathbb{E}\left[S_{h}^{t}\right], \forall t\right] \leq e^{\frac{-\epsilon^{2} \cdot N \cdot f_{h}}{10}} \leq \frac{1}{\gamma^{\beta}}
$$

Thus, because $\mathbb{E}\left[S_{h}^{t}\right] \geq f_{h} n_{h}$ for all $t$, we have

$$
\operatorname{Pr}\left[\left|V_{h} \cap \operatorname{prot}(V, E \backslash F, s)\right| \geq(1-\epsilon) f_{h} n_{h}\right] \geq \operatorname{Pr}\left[\exists t: S_{h}^{t}>(1-\epsilon) \mathbb{E}\left[S_{h}^{t}\right]\right] \geq 1-\frac{1}{\gamma^{\beta}}
$$

A union bound over all demographics would finally give

$$
\operatorname{Pr}\left[\left|V_{h} \cap \operatorname{prot}(V, E \backslash F, s)\right| \geq(1-\epsilon) f_{h} n_{h}, \forall h \in[\gamma]\right] \geq 1-\frac{1}{\gamma^{\beta-1}}
$$

By Lemma 3.2.4, in each run an edge $e$ gets removed with probability $x_{e}$. Hence, with a union bound over all runs, the probability that $e$ gets removed is at most $N x_{e}$. Therefore, the total expected cost of our algorithm is $N \sum_{e \in E} w_{e} x_{e}$, and since LP (3.3)-(3.6) is a valid relaxation of the problem, we immediately get the desired approximation ratio on expectation. By Markov's inequality we can further prove that with probability at most $\frac{1}{c}$, we get a final cut of cost more
than $c N \sum_{e \in E} w_{e} x_{e}$ for some constant $c>1$.
Thus, with constant probability our algorithm satisfies both the ratio of $O\left(\frac{\log \gamma}{\epsilon^{2} \min _{h} f_{h}}\right)$, and the $1-\epsilon$ approximate satisfaction of the demographic constraints (specifically we fail to satisfy both of the above with probability at most $1 / \gamma^{\beta-1}+1 / c$ ). Hence, repeating the whole process an expected logarithmic number of times, guarantees that we hit both targets deterministically.

By combining Theorem 3.2.8 and Lemma 3.1.3, our approach achieves the following.

Theorem 3.2.9. For any given constant $\epsilon>0$, we provide a $\left(O\left(\frac{\log n \log \gamma}{\epsilon^{2} \cdot \min _{h} f_{h}}\right), 1-\epsilon\right)$-bicriteria algorithm for DEMFAIRCUT, which also runs in expected polynomial time.

### 3.2.2.1 Hardness of DEMFAIRCUT with Arbitrary $\gamma$

Here we show that even in tree instances, DEmFAIRCuT with arbitrary $\gamma$ is hard. Specifically, we use an approximation preserving reduction from SET COVER.

SET Cover: We are given a universe of elements $U$ and $m$ sets $\left\{S_{1}, S_{2}, \ldots, S_{m}\right\}$, where $S_{i} \subseteq U$ for every $i \in[m]$. The goal is to find $C \subseteq[m]$, such that $\bigcup_{i \in C} S_{i}=U$ and $|C|$ is minimized.

Theorem 3.2.10 ([124]). It is NP-hard to approximate Set Cover instances of universe size $n$ and $m \leq \operatorname{poly}(n)$ sets within a factor better than $\ln n$.

Theorem 3.2.11. It is $N P$-hard to approximate DEMFAIRCUT with arbitrary $\gamma$ on tree instances within a factor better than $\ln \gamma$.

Proof. Suppose that we are given an instance of Set Cover. We create an instance of DemFAIRCUT as follows. For every set $S_{i}$ we create a vertex $v_{i}$. For every element $e \in U$ we create a demographic group $V_{e}=\left\{v_{i} \mid e \in S_{i}\right\}$. We set the covering requirement of the group $V_{e}$ to
be $1 /\left|V_{e}\right|$, i.e., we want our solution to protect at least $\left|V_{e}\right| \cdot\left(1 /\left|V_{e}\right|\right)=1$ vertex from each $V_{e}$. Finally, we add the designated vertex $s$ to the graph, and create edges $\left(s, v_{i}\right)$ for every $v_{i}$. Note that the resulting graph is a tree.

Now consider the optimal SET Cover solution $C^{*}$. We claim that the set of edges $\left\{\left(s, v_{i}\right) \mid i \in\right.$ $\left.C^{*}\right\}$ is a feasible solution for the constructed instance of DEMFAIRCUT. Take any demographic $V_{e}$ for $e \in U$. Because $C^{*}$ is a feasible SEt Cover solution, it contains at least one $S_{j}$ with $e \in S_{j}$. Therefore, we are going to include the edge $\left(s, v_{j}\right)$ to our graph solution, and the vertex $v_{j}$ from the group $V_{e}$ is going to be protected. Finally, see that $\left|C^{*}\right|=\left|\left\{\left(s, v_{i}\right) \mid i \in C^{*}\right\}\right|$, and hence the cost of the optimal solution for the DEmFairCut instance, say $F^{*}$, is at most $\left|C^{*}\right|$.

Now we argue that any solution $F$ to the DEmFairCut instance yields a feasible solution $C_{F}$ for the Set Cover instance with $|F|=\left|C_{F}\right|$. Simply take $C_{F}=\left\{i \in[m] \mid\left(s, v_{i}\right) \in F\right\}$. It is clear that $|F|=\left|C_{F}\right|$. Now consider each $e \in U$. Since $F$ is feasible for DEmFairCut, at least one vertex $v_{i} \in V_{e}$ will be separated from $s$, and thus $\left(s, v_{i}\right) \in F$. Hence for that vertex $v_{i}$ we have $e \in S_{i}$ by construction. Therefore, $e$ is covered by $C_{F}$.

Suppose now that for some $\epsilon>0$ we have an $(1-\epsilon) \ln \gamma$-approximation algorithm for DemFairCut on trees. Then given an instance of Set Cover, we first construct the instance of DemFaircut given by the above reduction and then run the given algorithm on that instance to get a solution $F$. Then, as discussed, we construct the corresponding Set Cover solution $C_{F}$, with $|F|=\left|C_{F}\right|$. By all the previous arguments we have

$$
\left|C_{F}\right|=|F| \leq((1-\epsilon) \ln \gamma)\left|F^{*}\right| \leq((1-\epsilon) \ln |U|)\left|C^{*}\right|
$$

This contradicts Theorem 3.2.10.

At a high-level, the previous theorem says that the best we can achieve for DEMFAIRCUT in trees is an approximation ratio of $\Omega(\log \gamma)$. Trivially this implies the following corollary.

Corollary 3.2.12. Unless $P=N P$, the best approximation ratio we can achieve for general instances of DEMFAIRCUT with arbitrary $\gamma$ is $\Omega(\log \gamma)$.

### 3.3 Solving IndFairCut

The purpose of this section is to provide an algorithm for InDFAIRCuT. We begin by giving a dynamic programming bicriteria algorithm for AUXCUT on tree instances, which according to Lemma 3.1.5 implies an algorithm for AUXCUT in general graphs. Subsequently, we show how the general graph algorithm can be incorporated in the round-or-cut framework of Anegg et al. [16], and in this way we get as our final result a $O(\log n)$-approximation for IndFAIRCUT.

At this point, we have to mention that the LP-based approach of Section 3.2.2 can also be applied here (by adding the extra constraint $y_{v} \geq p_{v}$ in LP (3.3)-(3.6)), yielding the same approximation ratio of $O(\log n)$. However, such an approach would unavoidably lead to a bicriteria algorithm, since it will produce a solution that saves at least $(1-\epsilon) T$ vertices. On the other hand, the algorithm we present in what follows is a true approximation for InDFAIRCUT.

### 3.3.1 A $(1,1, O(\log n)+\epsilon)$-Bicriteria Algorithm for AUXCUT

Suppose that we have an instance $\mathcal{I}=(V, E, B, T, s, w, a)$ of AuxCut. Given the reduction of Lemma 3.1.5, we can focus on the case of $G=(V, E)$ being a tree. We now present a dynamic programming bicriteria algorithm for AUXCUT in trees.

Without loss of generality, we can assume that the tree is rooted at $s$ and is binary (see

Lemma 15.18 from [122]). Our algorithm tries to find a cut $F \subseteq E$ that minimizes $a(V \backslash$ $\operatorname{prot}(V, E \backslash F, s))$ subject to $w(F) \leq B$ and $|\operatorname{prot}(V, E \backslash F, s)| \geq T$. Note that when we can compute a solution of optimal value to this minimization problem, minimizing $a(V \backslash \operatorname{prot}(V, E \backslash$ $F, s)$ ) is equivalent to maximizing $a(\operatorname{prot}(V, E \backslash F, s))$. Therefore, the version of the problem we solve here is equivalent to the definition of AUXCUT as given in Section 3.1.

Our approach relies on a table $A$. For every $v \in V$ let $T_{v} \subseteq V$ and $E_{v} \subseteq E$ be the vertices and the edges of the subtree that is rooted at $v$ (with $v$ included in $T_{v}$ ). Then, the entry $A[v, W, k]$ would represent the minimum possible $a\left(T_{v} \backslash \operatorname{prot}\left(T_{v}, E_{v} \backslash F_{v}, v\right)\right)$, for any cut $F_{v} \subseteq E_{v}$ with $w\left(F_{v}\right)=W$ and $\left|T_{v} \backslash \operatorname{prot}\left(T_{v}, E_{v} \backslash F_{v}, v\right)\right|=k$ (see that the vertices of $T_{v}$ connected to $v$ in this cut are those in $T_{v} \backslash \operatorname{prot}\left(T_{v}, E_{v} \backslash F_{v}, v\right)$ ). Let also $v_{r}$ be the right child of $v$, and let $v_{\ell}$ be the left child of $v$. The optimal solution of $\mathcal{I}$ either cuts none of the edges from $v$ to its children, just the left edge, just the right edge, or both edges. So we just have to set $A[v, W, k]$ to the minimum of the following four cases:

1. $\min \left\{A\left[v_{\ell}, W_{\ell}, k_{\ell}\right]+A\left[v_{r}, W_{r}, k_{r}\right]+a_{v}: W_{\ell}+W_{r}=W\right.$ and $\left.k_{\ell}+k_{r}+1=k\right\}$
2. $A\left[v_{r}, W-w_{\left(v, v_{\ell}\right)}, k-1\right]+a_{v}$ if $W \geq w_{\left(v, v_{\ell}\right)}$ and $k>1,+\infty$ otherwise
3. $A\left[v_{\ell}, W-w_{\left(v, v_{r}\right)}, k-1\right]+a_{v}$ if $W \geq w_{\left(v, v_{r}\right)}$ and $k>1,+\infty$ otherwise
4. $a_{v}$ if $w_{\left(v, v_{\ell}\right)}+w_{\left(v, v_{r}\right)}=W$ and $k=1,+\infty$ otherwise

The first case above corresponds to cutting neither of the edges $\left(v, v_{r}\right),\left(v, v_{\ell}\right)$, the second to cutting only $\left(v, v_{\ell}\right)$, the third to cutting only $\left(v, v_{r}\right)$, and the fourth to cutting both.

To fill in $A$, we begin by initializing $A[v, 0,1]=a_{v}$ for all leaves $v$ of the tree, and all other entries to $+\infty$. Then we proceed by filling the table bottom-up. Assuming that the edge weights
are integers, we see that $A$ has $n^{2} B$ entries, and in order to fill each of them, we need access to at most $2 n B$ other entries. Hence, in total our approach requires $O\left(n^{3} B^{2}\right)$ time. Finally, in order to find the optimal cut, we look for the minimum entry $A[s, W, k]$, such that $W \leq B$ and $k \leq n-T$.

Corollary 3.3.1. When the edge weights are integers and $B=\operatorname{poly}(n)$, we can efficiently find an optimal solution of AUXCUT in tree instances.

To make sure the edge weights are integers and $B$ is polynomially bounded, we use a standard discretization trick before running the dynamic program [13]. Specifically, for any $\epsilon>0$, let $\lambda=\frac{\lceil m / \epsilon]}{B}$, where $m=|E|$. Then for each edge $e \in E$ create a new weight $w_{e}^{\prime}=$ $\left\lfloor\lambda w_{e}\right\rfloor$. Also, set $B^{\prime}=\lambda W=\lceil m / \epsilon\rceil$. Notice now that all new edge weights are integers and that $B^{\prime}$ is polynomial in $n$. Further, using these new values we create a new instance $\mathcal{I}^{\prime}=$ ( $V, E, B^{\prime}, T, s, w^{\prime}, a$ ) of AuxCut. It is easy to see that if there is a solution of edge-cost $B$ for $\mathcal{I}$, then this solution has edge-cost $B^{\prime}$ in $\mathcal{I}^{\prime}$. In addition, for every solution of $\mathcal{I}^{\prime}$ whose edge-cost is at most $B^{\prime}$, its edge-cost in $\mathcal{I}$ is at most $(1+\epsilon) B$. Combining this with Corollary 3.3.1 gives:

Corollary 3.3.2. Our approach provides a $(1,1,1+\epsilon)$-bicriteria for AUXCUT in tree instances.

Theorem 3.3.3. Our approach provides a $(1,1, O(\log n)+\epsilon)$-bicriteria algorithm for AUXCUT.

### 3.3.2 A Round-or-Cut Solution for IndFAIRCUT

Suppose we are given an instance $\mathcal{I}=(V, E, T, s, w, \vec{p})$ of IndFAIRCUT with optimal value $O P T_{\mathcal{I}}$. For any value $B \geq 0$, let $\mathcal{F}(B)=\{F \subseteq E: w(F) \leq B$ and $|\operatorname{prot}(V, E \backslash F, s)| \geq$ $T\}$. In the rest of the section we demonstrate a process, which given $\mathcal{I}$, a target value $B$ and $\epsilon>0$, operates as follows. It either returns an efficiently-sampleable distribution $\mathcal{D}$ over the cuts
in the set $\mathcal{F}((O(\log n)+\epsilon) B)$ such that $\operatorname{Pr}_{F \sim \mathcal{D}}[v \in \operatorname{prot}(V, E \backslash F, s)] \geq p_{v}$ for every $v \in V \backslash\{s\}$, or returns "INFEASIBLE". If the latter happens, then it is guaranteed that $B<O P T_{\mathcal{I}}$.

Using the above process in a bisection search with step $(1+\epsilon)$ over the range $[0, w(E)]$, we can efficiently compute a value $B^{\prime} \leq(1+\epsilon) O P T_{\mathcal{I}}$, such that the process will not return "INFEASIBLE" for $B^{\prime}$. This actually yields an efficiently-sampleable distribution over $\mathcal{F}\left((O(\log n)+\epsilon) B^{\prime}\right)$ that satisfies the stochastic constraints for all vertices. Hence, we get:

Theorem 3.3.4. For any $\epsilon>0$ and instance $\mathcal{I}$ with optimal value $O P T_{\mathcal{I}}$, we construct an efficiently sampleable distribution $\mathcal{D}$ over $\mathcal{F}\left(O(\log n+\epsilon) O P T_{\mathcal{I}}\right)$, such that $\operatorname{Pr}_{F \sim \mathcal{D}}[v \in \operatorname{prot}(V, E \backslash$ $F, s)] \geq p_{v}$ for every $v \in V \backslash\{s\}$. Moreover, the runtime of our approach is $\operatorname{poly}\left(n^{1 / \epsilon}\right)$.

Therefore, since for our final result the aforementioned process is all that is required, we start describing its details. Notice now that for a given target value $B$, we are basically interested in verifying whether or not there is a feasible solution to $\mathcal{I}$ with edge-cost at most $B$. Hence, consider the following exponential-sized linear program, which we call $\operatorname{PLP}(B)$.

$$
\begin{aligned}
& \max \sum_{v \in V \backslash\{s\}} p_{v} \cdot y_{v}-\mu \\
& \sum_{v \in \operatorname{prot}(V, E \backslash F, s)} y_{v} \leq \mu \quad \forall F \in \mathcal{F}(B) \\
& 0 \leq y_{v} \quad \forall v \in V
\end{aligned}
$$

$$
\sum_{F \in \mathcal{F}(B)} x_{F}=1
$$

$$
0 \leq x_{F} \leq 1 \quad \forall F \in \mathcal{F}(B)
$$

If we interpret $x_{F}$ as the probability of choosing the solution $F$, we see that $B$ yields a feasible solution iff $\operatorname{PLP}(B)$ is feasible. This is because the first LP constraint captures the
fairness requirements, and the second LP constraint the fact that the resulting solution should be a distribution over $\mathcal{F}(B)$. In addition, if $\operatorname{PLP}(B)$ is feasible, then there are only $n$ values $x_{F}$ with $x_{F}>0$ (see Lemma 9 in [125]), and hence the resulting distribution is efficiently-sampleable. Another important observation is that if $\operatorname{PLP}(B)$ is feasible, then clearly its optimal value is 0 .

However, since solving $\operatorname{PLP}(B)$ is not doable in polynomial time, we focus on its dual, which we call $\operatorname{DLP}(B)$ and we present next to the primal LP.

Here note that $\operatorname{DLP}(B)$ is always feasible (e.g., set all variables to 0 ), and by LP duality $\operatorname{DLP}(B)$ has an optimal value of 0 iff $\operatorname{PLP}(B)$ is feasible. Moreover, it is easy to see that $\operatorname{DLP}(B)$ is scale-invariant. In other words, if it has a feasible solution $\left(y^{\prime}, \mu^{\prime}\right)$ with strictly positive objective value, then we can reason that $\operatorname{DLP}(B)$ is unbounded, because $\left(t y^{\prime}, t \mu^{\prime}\right)$ will also be feasible for any $t>0$. Consider now the following polytope that contains all feasible solutions of $\operatorname{DLP}(B)$ of objective value at least 1 .

$$
Q(B)=\left\{(y, \mu) \in \mathbb{R}_{\geq 0}^{n-1} \times \mathbb{R}: \sum_{v \in V \backslash\{s\}} p_{v} y_{v} \geq \mu+1 \wedge y(\operatorname{prot}(V, E \backslash F, s)) \leq \mu, \forall F \in \mathcal{F}(B)\right\}
$$

Based on the previous discussion we make the following very crucial observation.

Observation 3.3.5. $\operatorname{PLP}(B)$ is feasible iff $Q(B)=\emptyset$.

Using the algorithm of Section 3.3.1 we prove the following vital theorem.

Theorem 3.3.6. There exists a poly-time algorithm that given a point $(y, \mu) \in \mathbb{R}_{\geq 0}^{n-1} \times \mathbb{R}$ satisfying $\sum_{v \in V \backslash\{s\}} p_{v} \cdot y_{v} \geq \mu+1$, it either verifies that $(y, \mu) \in Q(B)$, or outputs a set $F \in \mathcal{F}((O(\log n)+$ $\epsilon) B)$ such that $\sum_{v \in \operatorname{prot}(V, E \backslash F, s)} y_{v}>\mu$.

Proof. We begin by constructing an instance $\mathcal{I}_{\text {aux }}=(V, E, B, T, s, w, y)$ of AUXCUT, where the vertex weights correspond to the $y$ values. Then, we run the algorithm of Section 3.3.1 on $\mathcal{I}_{\text {aux }}$.

Suppose now that $F \subseteq E$ is the solution returned by the algorithm, for which by Theorem 3.3.3 we have $w(F) \leq(O(\log n)+\epsilon) B$ and $|\operatorname{prot}(V, E \backslash F, s)| \geq T$. If $y(\operatorname{prot}(V, E \backslash F, s))>\mu$, then we return $F$ as our answer, because we are guaranteed to have $F \in \mathcal{F}((O(\log n)+\epsilon) B)$. If on the other hand $y(\operatorname{prot}(V, E \backslash F, s)) \leq \mu$, then all $F^{\prime} \in \mathcal{F}(B)$ have $y\left(\operatorname{prot}\left(V, E \backslash F^{\prime}, s\right)\right) \leq \mu$, because the properties of the Section 3.3.1 algorithm ensure that $y(\operatorname{prot}(V, E \backslash F, s)) \geq y(\operatorname{prot}(V, E \backslash$ $\left.F^{\prime}, s\right)$ ). The latter immediately indicates that $(y, \mu) \in Q(B)$.

Given the existence of an algorithm like the one described in Theorem 3.3.6, Anegg et al. [16] prove that with a round-or-cut approach we can either show that $Q(B) \neq \emptyset$ or that $Q((O(\log n)+$ $\epsilon) B)=\emptyset$. If $Q(B) \neq \emptyset$, then by Observation 3.3.5 we can infer $B<O P T_{\mathcal{I}^{\prime}}$ and return "INFEASIBLE". If on the other hand $Q((O(\log n)+\epsilon) B)=\emptyset$, then again by Observation 3.3.5 we know that $\operatorname{PLP}((O(\log n)+\epsilon) B)$ is feasible. Furthermore, in the latter case the framework of Anegg et al. [16] provides a set $\mathcal{F}^{\prime} \subseteq \mathcal{F}((O(\log n)+\epsilon) B)$ with polynomial size, for which the following (poly-sized) LP is feasible.
$\min 0$

$$
\begin{array}{cc}
\sum_{\substack{F \in \mathcal{F}^{\prime} \\
v \in \operatorname{prot}(V, E \backslash F, s)}} x_{F} \geq p_{v} & \forall v \in V \backslash\{s\} \\
\sum_{F \in \mathcal{F}^{\prime}} x_{F}=1 & \\
0 \leq x_{F} \leq 1 & \forall F \in \mathcal{F}^{\prime}
\end{array}
$$

Finally, since the above can be efficiently solved, we obtain an efficiently-sampleable distribution $\mathcal{D}$ over $\mathcal{F}((O(\log n)+\epsilon) B)$, such that $\operatorname{Pr}_{F \sim \mathcal{D}}[v \in \operatorname{prot}(V, E \backslash F, s)] \geq p_{v}$ for all $v \in V \backslash\{s\}$.

## Chapter 4: Mitigating the Stochastic Spread of A Disease

### 4.1 MinInfEdge with Unit Edge-Costs and Uniform Probabilities

In this section we are going to consider a special case of MinInfEdge. Specifically, we assume that the edge costs of the network $G=(V, E)$ are all 1, i.e., $c_{e}=1$ for all $e \in E$. Moreover, we will work under the uniform transimission probability setting.

For a random graph $G(p)=(V, E(p))$ and any $F \subseteq E$, let $F(p)=F \cap E(p)$ be the random cut corresponding to $F$ in $G(p)$. Let also $c_{m i n}$ be the size of the smalletst cut in $G$. We are going to use a cut sparsification result of Karger [56].

Theorem 4.1.1. ([56]) Let $\epsilon=\sqrt{\frac{3(d+2)(\ln n)}{c_{m i n} \cdot p}}$ for some $d>0$. If $\epsilon \leq 1$ then with probability at least $1-O\left(1 / n^{d}\right)$, we have $\left||F(p)|-\mathbb{E}_{G(p)}[|F(p)|]\right| \leq \epsilon \mathbb{E}_{G(p)}[|F(p)|]$ for every $F \subseteq E$.

Observation 4.1.2. When $c_{\text {min }} \cdot p \geq 9 \ln n$, the statement of Theorem 4.1.1 holds with high probability, i.e., with probability at least $1-O(1 / n)$.

Observation 4.1.2 basically determines the regime where the results of this section hold. However, notice that $c_{\text {min }} \cdot p \geq 9 \ln n$ is actually a realistic assumption, since for most real-life scenarios the transmission probability will be some constant, and the size of the minimum cut in $G$ can very well be $\Omega(\ln n)$.

To tackle MinInfEdge in the current setting, we are going to reduce it to a problem
from [12], namely the Minimum-Size Bounded-Capacity Cut problem (MinSBCC). In this problem, we are given a graph $G=(V, E)$, a source vertex $s \in V$, and a budget $B$. We are then asked to find a set $F \subseteq E$ of at most at most $B$ edges, which minimizes the number of nodes in the same component as $s$ in $G_{F}=(V, E \backslash F)$, i.e., $\inf (V, E \backslash F, s)$; this problem can be interpreted as the dual of SB-MinCC. The main result of Hayrapetyan et al. [12] follows.

Theorem 4.1.3. For any $\lambda \in(0,1)$, there exists a polytime $\left(\frac{1}{\lambda}, \frac{1}{1-\lambda}\right)$-approximation algorithm for MinSBCC: it finds a cut of size at most $\frac{1}{\lambda} B$, in which the number of nodes in the same component as $s$ in the resulting subgraph is at most $\frac{1}{1-\lambda}$ times the value of the optimal solution with size $B$.

Our approach for solving MinInfEdge goes as follows. At first, we sample a graph $H=\left(V, E^{\prime}\right)$ from $G(p)$. Then, we create an instance of MinSBCC, where the graph under consideration is $H$, the source vertex is $s$, and the budget is $\gamma B p$ for a small constant $\gamma$ which we set later. Finally, we run the $\left(\frac{1}{\lambda}, \frac{1}{1-\lambda}\right)$-approximation of Hayrapetyan et al. [12] on the created instance of MinSBCC, and get a solution $F^{\prime} \subseteq E^{\prime}$. Let now $S$ be all the vertices that are in the same connected component as $s$ in $H_{F^{\prime}}=\left(V, E^{\prime} \backslash F^{\prime}\right)$. Our returned solution for the original instance of MinInfEdge is $\bar{F}=\{\{u, v\} \in E: u \in S, v \notin S\}$.

Lemma 4.1.4. When the assumption of Observation 4.1.2 holds, $|\bar{F}| \leq \frac{\gamma}{(1-\epsilon) \lambda} B$ with probability at least $1-O(1 / n)$, where $\epsilon$ is as in Theorem 4.1.1 and $\epsilon \in(0,1)$.

Proof. Notice that the vertices that are in the same connected as $s$ in $(V, E \backslash \bar{F})$, are exactly those that are connected to $s$ in $\left(V, E^{\prime} \backslash F^{\prime}\right)$. Therefore, the random cut corresponding to $\bar{F}$ in $G(p)$ is $F^{\prime}$, i.e., $F^{\prime}=\bar{F}(p)$. Hence, $\mathbb{E}_{G(p)}\left[F^{\prime}\right]=\mathbb{E}_{G(p)}[\bar{F}(p)]=p|\bar{F}|$. Thus, using Theorem 4.1.1, we
have that with probability at least $1-O(1 / n)$ :

$$
\left|\left|F^{\prime}\right|-p\right| \bar{F}||\leq \epsilon p| \bar{F}| \Longrightarrow\left|F^{\prime}\right| \geq(1-\epsilon) p|\bar{F}|
$$

Since $\left|F^{\prime}\right| \leq \frac{\gamma B p}{\lambda}$ (we ran the algorithm of Theorem 4.1.3 with budget $\gamma B p$ ), we get

$$
\frac{\gamma B p}{\lambda} \geq\left|F^{\prime}\right| \geq(1-\epsilon)|\bar{F}| p
$$

with probability at least $1-O\left(\frac{1}{n}\right)$. Rearranging terms implies $|\bar{F}| \leq \frac{\gamma}{(1-\epsilon) \lambda} B$.

Lemma 4.1.5. $|S| \leq \frac{\gamma}{1-\lambda} O P T$ with probability at least $1-\frac{2}{\gamma}$, where $O P T$ is the value of the optimal solution (the expected number of nodes infected).

Proof. Let $F^{*}$ denote the optimal solution (so $\left|F^{*}\right| \leq B$ ), and let $\hat{F}=F^{*} \cap E^{\prime}$ be a random variable denoting the edges of $F^{*}$ that are present in $E^{\prime}$. Let also $S_{\hat{F}}$ be the random variable denoting the nodes that are in the same connected component as $s$ in $\left(V, E^{\prime} \backslash \hat{F}\right)$. We say that there was a "success" in the process of sampling $H$ if the following two conditions are satisfied: 1) $|\hat{F}| \leq \gamma B p$ and 2) $\left|S_{\hat{F}}\right| \leq \gamma \cdot O P T$. If either condition is false we say that there was a "failure". Suppose that there was a success. Then the first condition implies that $\hat{F}$ was a feasible solution for the MinSBCC instance (since its size was within the given budget), and hence $|S| \leq$ $\frac{1}{1-\lambda}\left|S_{\hat{F}}\right|$. Then the second condition implies $|S| \leq \frac{\gamma}{1-\lambda} O P T$ as desired.

Finally, we need to show that the probability of success is at least $1-\frac{2}{\gamma}$, or equivalently that the probability of failure is at most $\frac{2}{\gamma}$. Clearly $\mathbb{E}_{G(p)}[|\hat{F}|]=p\left|F^{*}\right| \leq p B$, so by Markov's inequality $\operatorname{Pr}[|\hat{F}|>\gamma B p] \leq \frac{1}{\gamma}$. Similarly, $\mathbb{E}\left[\left|S_{\hat{F}}\right|\right]=O P T$ by definition of $O P T$, and so by Markov $\operatorname{Pr}\left[\left|S_{\hat{F}}\right|>\gamma \cdot O P T\right] \leq \frac{1}{\gamma}$. A union bound yields a failure probability of at most $\frac{2}{\gamma}$.

Theorem 4.1.6. There exists an $(O(1), O(1))$-approximation for MinInfEdge that works with high probability, as long as the assumption of Observation 4.1.2 holds.

Proof. If we set $\gamma$ to be a large enough constant, e.g., larger than 4 , then with probability at least $1 / 2-O(1 / n)$ we return a solution $\bar{F}$ which violates the budget by at most $O(1)$ (Lemma 4.1.4), and the size of the connected component in $(V, E \backslash \bar{F})$ which contains $s$ is at most $O(1) \cdot O P T$ (Lemma 4.1.5). Clearly this implies that $\mathbb{E}_{G(p)}[\inf (V, E(p) \backslash \bar{F}, s)]$ is also at most $O(1) \cdot O P T$. Thus, our algorithm gives the bounds in Theorem 4.1 .6 with constant probability. By repeating this process $O(\log n)$ times and taking the best single solution, this algorithm can be made to work with high probability (at least $1-O(1 / n)$ ).

### 4.2 The SAA Path-Dependent Framework for Arbitrary Networks

Consider a general instance of MinInfEdge. For a suitable number $N=\operatorname{poly}(n, m)$ that is going to be set later, we simulate the disease-percolation process on $G$ independently $N$ times. In other words, we independently sample $N$ graphs $G_{j}=\left(V, E_{j}\right)$, where $j \in[N]$ and $E_{j} \subseteq E$ is the subset of edges acquired in the $j^{\text {th }}$ simulation (or sample), when each edge $e \in E$ is retained with probability $p_{e}$. The heart of our approach is then to show how these "typical" samples $G_{j}$ can guide us towards computing a provably-good solution for our given percolation model.

We start by presenting the linear program LP (4.1)-(4.4). This LP models an "empirical" solution to the problem, when the diffusion process can only result in the graphs $G_{j}$, and each of these graphs materializes with probability $1 / N$. We use $\mathcal{P}\left(s, v, G_{j}\right)$ to denote the set of paths from $s$ to $v$ in the graph $G_{j}$, and $[k]$ to denote the set $\{1,2, \ldots, k\}$ for any positive integer $k$. For the integral version of our LP, $x_{e}$ is the indicator variable for removing edge $e$, and $y_{v j}$
the indicator for vertex $v$ not becoming reachable from $s$ in $G_{j}$ after our edge-removal. Then, constraint (4.2) makes sure that $v$ is disconnected from $s$ in $G_{j}$ iff for every path of $\mathcal{P}\left(s, v, G_{j}\right)$ at least one edge of the path has been removed. Constraint (4.3) captures the budget constraint, and the objective function (4.1) measures exactly the expected number of infections, when each $G_{j}$ appears with probability $1 / N$. Finally, in order to be able to efficiently solve the system, the $\{0,1\}$-variables are relaxed to lie in $[0,1]$.

$$
\begin{align*}
& \min \frac{1}{N} \sum_{j \in[N]} \sum_{v \in V}\left(1-y_{v j}\right) \text { such that }  \tag{4.1}\\
& \sum_{e \in P} x_{e} \geq y_{v j}, \forall j \in[N], \forall v \in V, \forall P \in \mathcal{P}\left(s, v, G_{j}\right)  \tag{4.2}\\
& \sum_{e \in E} c_{e} x_{e} \leq B  \tag{4.3}\\
& \quad x_{e}, y_{v j} \in[0,1], \text { for all } j \in[N], v \in V, e \in E \tag{4.4}
\end{align*}
$$

Our algorithm involves the following steps:

1. Solve LP (4.1)-(4.4), and let $x, y$ be the optimal fractional solution. This solution can be computed in polynomial time via the ellipsoid method, with a separation oracle that checks if the shortest-path distance from $s$ to $v$ in $G_{j}$ (with edge weights $x_{e}$ ) is less than $y_{v j}$ [126].
2. For a user-specified constant $\epsilon \in(0,1)$, define the following sets for the sake of analysis:

$$
S(j)=\left\{v \in V: y_{v j} \geq \epsilon\right\} \text { for every } j \in[N], \text { and } \mathcal{P}_{h i t}=\cup_{j} \cup_{v \in S(j)} \mathcal{P}\left(s, v, G_{j}\right)
$$

3. Let $F_{0}$ denote the set of edges which will constitute our returned solution. For some con-
stant $\gamma$ that will be defined later, put each edge $e \in E$ independently in $F_{0}$, with probability

$$
x_{e}^{\prime}=\min \left\{\frac{(\gamma+5) x_{e} \log n}{\epsilon}, 1\right\}
$$

For any fixed $F \subseteq E$, we define random variables $h\left(G_{j}, F\right)$ and $h(G, F)$, where the randomness here is over the choice of the $G_{j}$ 's, i.e., the randomness is of Type 3. Let $h\left(G_{j}, F\right)=$ $\inf \left(V, E_{j} \backslash F, s\right)$ and $h(G, F)=\frac{1}{N} \sum_{j=1}^{N} h\left(G_{j}, F\right)$; the former represents the number of infections in the $j$-th sample if $F$ are the edges to be removed, and the latter represents the average number of infections over the $N$ sampled graphs if again $F$ is the set of edges removed.

For the small user-defined constant $\epsilon>0$, we now choose $N=\frac{3 n}{\epsilon^{2}} \log \left(n^{2} \cdot 2^{m+1}\right)$ and present a simple concentration result in Lemma 4.2.1; note that for this choice we have $N=$ poly $(n, m)$ and hence our algorithm runs in polynomial time.

Lemma 4.2.1. For the chosen value $N=\frac{3 n}{\epsilon^{2}} \log \left(n^{2} \cdot 2^{m+1}\right)$, with probability at least $1-\frac{1}{n^{2}}$, we have $h(G, F) \in[(1-\epsilon) \mathbb{E}[h(G, F)],(1+\epsilon) \mathbb{E}[h(G, F)]]$ for all sets $F \subseteq E$. The expectation here is over randomness of Type 3, and specifically over the random sampling of the $N$ graphs $G_{j}$.

Proof. For a fixed set $F$, the quantities $h\left(G_{j}, F\right)$ are independent. Further, let $X_{j}=\frac{h\left(G_{j}, F\right)}{n} \in$ $[0,1]$ and $X=\sum_{j=1}^{N} X_{j}$. Note that $X=\frac{N}{n} h(G, F)$. Using a simple Chernoff bound yields:

$$
\begin{aligned}
\operatorname{Pr}[X \notin[(1-\epsilon) \mathbb{E}[X],(1+\epsilon) \mathbb{E}[X]]] & \leq 2 e^{-\frac{\epsilon^{2} \mathbb{E}[X]}{3}} \\
& =2 e^{-\frac{\epsilon^{2} N \cdot \mathbb{E}[h(G, F)]}{3 n}} \\
& \leq \frac{1}{n^{2} 2^{m}}
\end{aligned}
$$

To get the last inequality we use the definition of $N$, and the fact that $\mathbb{E}[h(G, F)] \geq 1$ (since there
is always at least one infection, namely the node $s$ ). Finally, since $X=\frac{N}{n} h(G, F)$, we have:

$$
\operatorname{Pr}[h(G, F) \notin[(1-\epsilon) \mathbb{E}[h(G, F)],(1+\epsilon) \mathbb{E}[h(G, F)]]] \leq \frac{1}{n^{2} 2^{m}}
$$

Because the number of subsets $F$ is $2^{m}$, a union bound over them concludes the proof.

Let $F^{*}=\arg \min _{F} \mathbb{E}[h(G, F)]$, where the expectation is over the random sampling of the graphs $G_{j}$ (Type 3 randomness). Since for every $F$ we have $\mathbb{E}\left[h\left(G_{j}, F\right)\right]=\mathbb{E}_{G(\vec{p})}[i n f(V, E(\vec{p}) \backslash$ $F, s)]$ for all $G_{j}$, and $\mathbb{E}[h(G, F)]=\frac{1}{N} \sum_{j} \mathbb{E}\left[h\left(G_{j}, F\right)\right]$, we see that $F^{*}$ is actually the optimal edge set for MinInfEdge. Also, we define the random variable $\hat{F}=\arg \min _{F} h(G, F)$, denoting the optimal integral solution of $\operatorname{LP}(4.1)-(4.4) ; \hat{F}$ is actually the optimal empirical solution for the sampled set of graphs. Recall now that $F_{0}$ is the subset of edges computed by our LP rounding algorithm, and recall the parameter $\Gamma$ from Section 1.4.4, indicating the expected number of paths in a randomly-drawn graph (with randomness being of types 1 and 3).

Proof. (Theorem 1.4.9) Showing the first part of the theorem is easy. Since each edge $e$ is removed (independently) with probability $x_{e}^{\prime}$, the expected cost of the removed edges is

$$
\begin{aligned}
\mathbb{E}\left[c\left(F_{0}\right)\right] & \leq \sum_{e} c_{e} x_{e}^{\prime} \\
& \leq \frac{(\gamma+5) \log n}{\epsilon} \sum_{e} c_{e} x_{e} \\
& \leq \frac{((\gamma+5) \log n) B}{\epsilon}
\end{aligned}
$$

where the last inequality follows from constraint (4.3). Next, we can assume w.l.o.g. that $B=1$. To do so, we first hard-wire $x_{e}=0$ for all edges $e$ with $c_{e}>B$, thus ignoring these edges in our
edge-removal problem. Then, we uniformly scale all remaining $c_{e}$ 's and the budget by a factor of $1 / B$. Using a standard Chernoff bound will give:

$$
\operatorname{Pr}\left[c\left(F_{0}\right) \geq(6(\gamma+5) \log n) / \epsilon\right] \leq O\left(1 / n^{\gamma}\right)
$$

We next prove the second part of the theorem. The event $\mathcal{A}$ that is a function of the randomness of types 1,2 , and 3 is the conjunction of the following three events:

- $\mathcal{A}_{1}$ : For each $P \in \mathcal{P}_{h i t}$, there exists an edge $e \in P$, such that $e \in F_{0}$.
- $\mathcal{A}_{2}: h\left(G, F^{*}\right) \leq(1+\epsilon) \mathbb{E}\left[h\left(G, F^{*}\right)\right]$.
- $\mathcal{A}_{3}: h\left(G, F_{0}\right) \geq(1-\epsilon) \mathbb{E}\left[h\left(G, F_{0}\right)\right]$.

We first show that $\mathbb{E}\left[\inf \left(V, E(\vec{p}) \backslash F_{0}, s\right) \mid \mathcal{A}\right] \leq(1+O(\epsilon)) O P T$, and then lower-bound $\operatorname{Pr}[\mathcal{A}]$.
Let us first condition on $\mathcal{A}$. Consider any $j \in[N]$. By $\mathcal{A}_{1}$ and the definition of the set $\mathcal{P}_{h i t}$, the only vertices in $\left(V, E_{j} \backslash F_{0}\right)$ that are reachable from $s$ can be those in $V \backslash S(j)$; these vertices are exactly the ones getting infected in the $j$-th sample. Further, by definition we have $y_{v j}<\epsilon$ for every $v \in V \backslash S(j)$. Therefore, the empirical number of infections over all the samples is:

$$
\begin{align*}
h\left(G, F_{0}\right) & \leq \frac{1}{N} \sum_{j \in[N]} \sum_{v \notin S(j)} 1 \\
& \leq \frac{1}{N} \sum_{j \in[N]} \sum_{v \notin S(j)} \frac{1-y_{v j}}{1-\epsilon} \\
& \leq \frac{h(G, \hat{F})}{1-\epsilon} \leq \frac{h\left(G, F^{*}\right)}{1-\epsilon} \tag{4.5}
\end{align*}
$$

The second inequality above follows because the LP value is a lower bound on $h(G, \hat{F})$, and the
last inequality follows since $\hat{F}$ minimizes $h(G, F)$. (4.5) and the definitions of $\mathcal{A}_{2}, \mathcal{A}_{3}$ yield

$$
\begin{aligned}
\mathbb{E}\left[h\left(G, F_{0}\right)\right] & \leq \frac{h\left(G, F_{0}\right)}{1-\epsilon} \\
& \leq \frac{h\left(G, F^{*}\right)}{(1-\epsilon)^{2}} \\
& \leq \frac{(1+\epsilon)}{(1-\epsilon)^{2}} \mathbb{E}\left[h\left(G, F^{*}\right)\right] \\
& =(1+O(\epsilon)) \mathbb{E}\left[h\left(G, F^{*}\right)\right]
\end{aligned}
$$

To conclude the proof we need to lower-bound $\operatorname{Pr}[\mathcal{A}]$. First, Lemma 4.2.1 shows that each of $\mathcal{A}_{2}$ and $\mathcal{A}_{3}$ holds with probability at least $1-1 / n^{2}$. Let us consider $\mathcal{A}_{1}$ next.

Let $\mathcal{B}$ be a random variable denoting the number of paths over all the samples $G_{j}$. Since $\Gamma$ is the expected number of paths in a single graph, linearity of expectation gives

$$
\mathbb{E}[\mathcal{B}]=\Gamma N=O\left(\frac{n^{3} \Gamma}{\epsilon^{2}}\right)
$$

since $m=O\left(n^{2}\right)$. Thus, by using Markov's inequality we have

$$
\operatorname{Pr}\left[\mathcal{B}=\Omega\left(\frac{n^{5} \Gamma}{\epsilon^{2}}\right)\right] \leq O\left(1 / n^{2}\right) \Longrightarrow \operatorname{Pr}\left[\mathcal{B}=O\left(\frac{n^{5} \Gamma}{\epsilon^{2}}\right)\right] \geq 1-O\left(1 / n^{2}\right)
$$

The randomness in the previous statements is of types 1 and 3 .
Consider now a path $P \in \mathcal{P}_{h i t}$. If there exists an $e \in P$ such that $x_{e}^{\prime}=1$, then this path is broken. Hence, assume that for all $e \in P$ we have $x_{e}^{\prime}<1$. By the definition of the paths in $\mathcal{P}_{\text {hit }}$ we also have $\sum_{e \in P} x_{e}^{\prime} \geq(\gamma+5) \log n$. Therefore, keeping in mind that we choose the edges of
$F_{0}$ independently, the probability that all edges of the path $P$ survive is at most

$$
\prod_{e \in P}\left(1-x_{e}^{\prime}\right) \leq e^{-\sum_{e \in P} x_{e}^{\prime}} \leq e^{-(\gamma+5) \log n} \leq n^{-(\gamma+5)}
$$

In the end, a union bound over all $P \in \mathcal{P}_{\text {hit }}$ gives:

$$
\operatorname{Pr}\left[\mathcal{A}_{1} \mid \mathcal{B}\right] \geq 1-\frac{\mathcal{B}}{n^{\gamma+5}}
$$

Combining everything gives $\operatorname{Pr}\left[\mathcal{A}_{1}\right] \geq\left(1-O\left(\frac{\Gamma}{\epsilon^{2} n^{\gamma}}\right)\right)\left(1-O\left(\frac{1}{n^{2}}\right)\right)=1-O\left(\frac{\Gamma}{\epsilon^{2} n^{\gamma}}\right)-O\left(\frac{1}{n^{2}}\right)$. Hence, putting down all the lower bounds for $\mathcal{A}_{1}, \mathcal{A}_{2}$ and $\mathcal{A}_{3}$ yields $\operatorname{Pr}[\mathcal{A}] \geq 1-O\left(\frac{\Gamma}{\epsilon^{2} n^{\gamma}}\right)-O\left(\frac{1}{n^{2}}\right)$.

Corollary 4.2.2. When $\Gamma=\operatorname{poly}(n), \mathbb{E}\left[\inf \left(V, E(\vec{p}) \backslash F_{0}, s\right)\right] \leq(1+O(\epsilon)+O(1 / n)) O P T$, where the randomness is with respect to Type 1 (if applicable), Type 2, and Type 3.

Proof. When $\Gamma=\operatorname{poly}(n)$, we set $\gamma$ large enough such that $O\left(\frac{\Gamma}{\epsilon^{2} n^{\gamma}}\right)=O\left(\frac{1}{n^{2}}\right)$. Using Theorem 1.4.9 we then have:

$$
\begin{align*}
\mathbb{E}\left[i n f\left(V, E(\vec{p}) \backslash F_{0}, s\right)\right] & =\mathbb{E}\left[\inf \left(V, E(\vec{p}) \backslash F_{0}, s\right) \mid \mathcal{A}\right] \operatorname{Pr}[\mathcal{A}]+\mathbb{E}\left[\inf \left(V, E(\vec{p}) \backslash F_{0}, s\right) \mid \overline{\mathcal{A}}\right] \operatorname{Pr}[\overline{\mathcal{A}}] \\
& \leq(1+O(\epsilon)) O P T+n O\left(1 / n^{2}\right)  \tag{4.6}\\
& \leq(1+O(\epsilon)+O(1 / n)) O P T
\end{align*}
$$

To get the first inequality we use the simply upper bound of $\mathbb{E}\left[\inf \left(V, E(\vec{p}) \backslash F_{0}, s\right) \mid \overline{\mathcal{A}}\right] \leq n$, and for the last one we use the fact that $1 \leq O P T$ ( $s$ is always infected).

### 4.3 Counting Paths in the Chung-Lu Random Graph Model

Recall the random graph model of Chung and Lu [22]. Here we are given vertices $V$, where each vertex $v \in V$ comes with a positive integer $w_{v}$ indicating its expected degree in the graph. For every pair of vertices $u$ and $v$, the edge $(u, v)$ is independently included in the graph with probability

$$
q_{u, v}=w_{u} w_{v} / \sum_{r \in V} w_{r}
$$

Furthermore, we consider a power-law model, in which $n_{i}$, the number of nodes of weight $i$, satisfies $n_{i}=\Theta\left(n / i^{\beta}\right)$, where $\beta>2$ is a given parameter. Finally, recall that $w_{\max }=\max _{v} w_{v}$, $w_{\text {min }}=\min _{v} w_{v}$, and a common assumption in this random graph setting is that $w_{\min }=O(1)$.

Take now any random graph $G=(V, E)$ that is produced by the above model. In that graph, we assume that a disease percolation process takes place, and this process is governed by some probability vector $\vec{p}$. We are interested in bounding the expected number of paths $\Gamma$ in $G(\vec{p})$, where the randomness of $\Gamma$ is obviously of both Types 1 and 3 . To do so, we start by analyzing the expected number of paths of length $k$ in $G$, where the randomness here is only of Type 1 . In what follows, we are using $\ell_{k}$ to denote the latter quantity.

Our first result is showing that when $\beta>3$, we have $\ell_{k} \leq \operatorname{poly}\left(n, 2^{k}\right)$. Furthermore, if $p_{e} \leq c_{0}$ for all $e \in E$, where $c_{0}$ is a universal positive constant, we demonstrate how to utilize the bound on $\ell_{k}$ and eventually give a polynomial bound on $\Gamma$.

In addition, when $\beta<3$ we provide a negative result, indicating that our SAA framework from Section 4.2 cannot be utilized for this case, as no polynomial bound on $\Gamma$ can be guaranteed.

By an abuse of notation, we will let $m$ denote the expected number (not actual number) of
edges in the graph $G$. Trivially, $m=\sum_{v \in V} w_{v} / 2$. Since $\beta>2, m$ can also be expressed as:

$$
\begin{align*}
m & =\Theta\left(\sum_{i} i \cdot n_{i}\right) \\
& =\Theta\left(\int_{w_{\min }}^{w_{\max }} \frac{n}{z^{\beta-1}} d z\right)=\Theta\left(\frac{n}{w_{\min }^{\beta-2}}\right) \tag{4.7}
\end{align*}
$$

The following lemma is required for counting paths.

Lemma 4.3.1. Fix some length $k$, and suppose that we are given a positive integer $D \geq w_{\text {min }}$.
Let $S(D, k) \doteq\left\{\left(a\left(w_{\text {min }}\right), a\left(w_{\text {min }}+1\right), \ldots, a(D)\right):\left(\forall i, a(i) \in \mathbb{Z}_{\geq 0}\right) \text { and } \sum_{i} a(i)=k\right\}^{1}$. Then,

$$
\ell_{k} \leq n \cdot\left(\frac{2^{k} k!}{m^{k}}\right) \cdot \sum_{\mathbf{a} \in S\left(w_{\max }, k\right)} \prod_{i=w_{\min }}^{w_{\max }}\left(\binom{n_{i}}{a(i)} \cdot i^{2 a(i)}\right) .
$$

Proof. We say that a vertex $v$ is in class $i$ if $w_{v}=i$. Fix now a vertex $v_{0}$. We will next upper bound how many different paths of length $k$ can start from $v_{0}$. We can construct such a possible path $P=\left(v_{0}, u_{1}, \ldots, u_{k}\right)$ as follows:

1. Pick a vector $\mathbf{a}=\left(a\left(w_{\min }\right), a\left(w_{\min }+1\right), \ldots, a\left(w_{\max }\right)\right)$ from $S\left(w_{\max }, k\right)$. This will give us the selection of how many vertices from each degree class we should pick, such that in total we have chosen $k$ vertices for $P$.
2. For the chosen a, pick $a(i)$ vertices from each degree class $i$, where $w_{\min } \leq i \leq w_{\max }$. This is possible only if $a(i) \leq n_{i}$ for each class $i$. However, since we are only computing an upper bound, we will assume that such a selection is always possible. Notice that there may be some additional double counting, because we may end up choosing $v_{0}$ again.
[^5]Nonetheless, as we are only concerned with an upper bound, we will again permit such "unecessary" cases in our counting.
3. Choose the positions of the chosen $k$ vertices among the indices $\{1,2, \ldots, k\}$ of the path $P$ to be constructed ( $k$ ! possibilities).

Overall, based on the 3 cases above, the number of paths starting from $v_{0}$ can be at most:

$$
\begin{equation*}
k!\sum_{\mathbf{a} \in S\left(w_{\max }, k\right)} \prod_{i=w_{\min }}^{w_{\max }}\binom{n_{i}}{a(i)} \tag{4.8}
\end{equation*}
$$

Let $d_{0}$ be the class of $v_{0}$. Suppose we complete the three steps above and let $\left(d_{0}, d_{1}, \ldots, d_{k}\right)$ be the ordered degree sequence obtained for the vertices in $P$, when the chosen vector was a. The probability of such a path materializing in the edge selection phase is

$$
\begin{equation*}
\prod_{i=0}^{k-1}\left(\frac{d_{i} d_{i+1}}{\sum_{v} w_{v}}\right)=\frac{2^{k}}{m^{k}} \prod_{i=w_{\min }}^{w_{\max }} i^{2 a(i)} \tag{4.9}
\end{equation*}
$$

Combining (4.8) and (4.9), the expected number of length $k$ paths starting at $v_{0}$ is at most

$$
\begin{equation*}
\left(\frac{2^{k} k!}{m^{k}}\right) \cdot \sum_{\mathbf{a} \in S\left(w_{\max }, k\right)} \prod_{i=w_{\min }}^{w_{\max }}\left(\binom{n_{i}}{a(i)} \cdot i^{2 a(i)}\right) \tag{4.10}
\end{equation*}
$$

Summing this bound over all possible starting vertices results in the claim of the Lemma.

### 4.3.1 A Positive Result When $\beta>3$

We begin this section with a couple of important technical lemmas, and then move on to our final result regarding the expected number of paths in a randomly drawn graph.

Lemma 4.3.2. Let $S(D, k)$ be as in Lemma 4.3.1, and $N(D, k) \doteq \sum_{\mathbf{a} \in S(D, k)} \prod_{i=w_{\text {min }}}^{D} \frac{1}{i^{c_{1} a(i) \cdot a(i)!}}$ for some constant $c_{1}>1$. Then $N(D, k) \leq \frac{1}{k!} \cdot \prod_{i=w_{m i n}+1}^{D}\left(1+\frac{1}{i^{c_{1}}}\right)^{k}$.

Proof. We prove by induction on $D$ the stronger statement (A), which will imply the Lemma.

$$
\begin{equation*}
\forall k, N(D, k) \leq \frac{1}{k!} \cdot \prod_{i=w_{\min }+1}^{D}\left(1+\frac{1}{i^{c_{1}}}\right)^{k} \tag{A}
\end{equation*}
$$

The base case $D=w_{\min }$ is easy; notice that $S\left(w_{\min }, k\right)=\{(k)\}$, and hence we have $N\left(w_{\min }, k\right)=\frac{1}{w_{m i n}^{c_{1} k} \cdot k!} \leq \frac{1}{k!}$. The inequality above follows since $c_{1}>1$.

We complete the proof by strong induction. Suppose $D>w_{\text {min }}$. Elementary calculations and the definition of $N(\cdot, \cdot)$ reveal the following recurrence when $D>w_{\min }$

$$
\begin{equation*}
N(D, k)=\sum_{j=0}^{k}\left(N(D-1, k-j) \cdot \frac{1}{D^{c_{1} j} \cdot j!}\right) \tag{4.11}
\end{equation*}
$$

Recurrence (4.11) and the induction hypothesis yield

$$
\begin{aligned}
N(D, k) & \leq \sum_{j=0}^{k}\left(\frac{1}{D^{c_{1} j} \cdot j!} \cdot \frac{1}{(k-j)!} \prod_{i=w_{\min }+1}^{D-1}\left(1+\frac{1}{i^{c_{1}}}\right)^{k-j}\right) \\
& \leq \sum_{j=0}^{k}\left(\frac{1}{D^{c_{1} j} \cdot j!} \cdot \frac{1}{(k-j)!} \prod_{i=w_{\min }+1}^{D-1}\left(1+\frac{1}{i^{c_{1}}}\right)^{k}\right) \\
& =\left(\prod_{i=w_{\min }+1}^{D-1}\left(1+\frac{1}{i^{c_{1}}}\right)^{k}\right) \sum_{j=0}^{k}\left(\frac{1}{D^{c_{1} j} \cdot j!} \cdot \frac{1}{(k-j)!}\right) \\
& =\frac{1}{k!} \cdot\left(\prod_{i=w_{\min }+1}^{D-1}\left(1+\frac{1}{i^{c_{1}}}\right)^{k}\right) \cdot \sum_{j=0}^{k}\left(\binom{k}{j} \cdot \frac{1}{D^{c_{1} j}}\right) \\
& =\frac{1}{k!} \cdot\left(\prod_{i=w_{\min }+1}^{D}\left(1+\frac{1}{i^{c_{1}}}\right)^{k}\right)
\end{aligned}
$$

The last inequality above follows from the binomial sum $\left.\sum_{j=0}^{k}\binom{k}{j} \cdot \frac{1}{D^{c_{1 j}}}\right)=\left(1+1 / D^{c_{1}}\right)^{k}$.
Lemma 4.3.3. Suppose $\beta=2+c_{1}$ for some constant $c_{1}>1$. Then, for all $k, \ell_{k} \leq \operatorname{poly}\left(n, 2^{k}\right)$.

Proof. Before we proceed to our main arguments, we make some useful observations and give a bit more notation. At first, using (4.7) and the assumption that $w_{\text {min }}=O(1)$, we see that $\frac{n}{m}=O(1)$. Furthermore, because $n_{i}=\Theta\left(\frac{n}{i^{\beta}}\right)$ for every $i \in\left[w_{\text {min }}, w_{\text {max }}\right]$, let $\lambda$ be a universal constant such that $n_{i} \leq \frac{\lambda n}{i^{\beta}}$ for every $i$. Using Lemma 4.3.1 we get

$$
\begin{aligned}
\ell_{k} & \leq n \cdot\left(\frac{2^{k} k!}{m^{k}}\right) \cdot \sum_{\mathbf{a} \in S\left(w_{\max }, k\right)} \prod_{i=w_{\min }}^{w_{\max }}\left(\frac{n_{i}^{a(i)}}{a(i)!} \cdot i^{2 a(i)}\right) \\
& \leq n \cdot\left(\frac{2^{k} k!}{m^{k}}\right) \cdot \sum_{\mathbf{a} \in S\left(w_{\max }, k\right)} \prod_{i=w_{\min }}^{w_{\max }}\left(\frac{\left(\lambda n / i^{2+c_{1}}\right)^{a(i)}}{a(i)!} \cdot i^{2 a(i)}\right) \\
& \leq n \cdot k!\cdot\left(\frac{\lambda n}{m}\right)^{k} \cdot \sum_{\mathbf{a} \in S\left(w_{\max }, k\right)} \prod_{i=w_{\min }}^{w_{\max }}\left(\frac{\left(1 / i^{2+c_{1}}\right)^{a(i)}}{a(i)!} \cdot i^{2 a(i)}\right) \\
& =n \cdot k!\cdot\left(\frac{\lambda n}{m}\right)^{k} \cdot \sum_{\mathbf{a} \in S\left(w_{\max }, k\right)} \prod_{i=w_{\min }}^{w_{\max }} \frac{1}{i^{c_{1} a(i)} \cdot a(i)!} \\
& =\operatorname{poly}\left(n, 2^{k}\right) \cdot k!\cdot N\left(w_{\max }, k\right) .
\end{aligned}
$$

Using the bound on $N(D, k)$ from Lemma 4.3.2, we have

$$
\begin{aligned}
\ell_{k} & \leq \operatorname{poly}\left(n, 2^{k}\right) \cdot \prod_{i=w_{\min }+1}^{\infty}\left(1+\frac{1}{i^{c_{1}}}\right)^{k} \\
& \leq \operatorname{poly}\left(n, 2^{k}\right) \cdot \prod_{i=w_{\text {min }}+1}^{\infty} e^{\frac{k}{i^{\frac{k}{c 1}}}} \\
& =\operatorname{poly}\left(n, 2^{k}\right) \cdot e^{k \cdot \sum_{i>w_{\text {min }}} \frac{1}{i^{c 1}}} \\
& \leq \operatorname{poly}\left(n, 2^{k}\right)
\end{aligned}
$$

The last inequality follows because $\sum_{i>w_{\text {min }}}\left(1 / i^{c_{1}}\right)=O(1)$ when $c_{1}>1$.

Corollary 4.3.4. Let $G$ be a graph drawn from the Chung-Lu distribution with power law weights, with parameter $\beta=2+c_{1}$ for some constant $c_{1}>1$. Then there is a constant $c_{0}>0$ that depends only on $c_{1}$, such that the following holds: if the probability $p_{e}$ of retaining edge e during the disease percolation process satisfies $p_{e} \leq c_{0}$ for any $e$, then the expected number $\Gamma$ of paths in $G(\vec{p})$ is upper-bounded by poly $(n)$. (This expectation is over randomness of types 1 and 3.)

Proof. From Lemma 4.3.3, we have $\ell_{k} \leq \operatorname{poly}\left(n, 2^{k}\right)$; let $C$ be a constant such that $\ell_{k} \leq n^{C} 2^{C k}$ for all $k$. We choose $c_{0}=2^{-C}$. Then, when $p_{e} \leq c_{0}$ for every $e$, the probability that a given path of length $k$ survives in $G(\vec{p})$ is at most $c_{0}^{k}$. Therefore, the expected number of paths in $G(\vec{p})$ is

$$
\Gamma \leq \sum_{k} \ell_{k} c_{0}^{k} \leq \sum_{k} n^{C}\left(c_{0} 2^{C}\right)^{k} \leq n^{C+1}
$$

Combining Corollaries 4.3.4 and 4.2.2, we get a bicriteria approximation for MININF-CL.

### 4.3.2 A Negative Result When $\beta<3$

We now consider the case $\beta<3$ and show an interesting contrast to Lemma 4.3.3.

Lemma 4.3.5. When $\beta=2+c_{0}$ for some $c_{0}<1$, there might exist $k$ with $\ell_{k}=\omega\left(\operatorname{poly}\left(n, 2^{k}\right)\right)$.

Proof. In the proof of Lemma 4.3.1 we gave an upper bound for $\ell_{k}$. However, the double counting or the unnecessary cases we involved in our counting can only account for low-order terms. In other words, we can assume

$$
\begin{equation*}
\ell_{k}=\Theta\left(n \cdot\left(\frac{2^{k} k!}{m^{k}}\right) \cdot \sum_{\mathbf{a} \in S\left(w_{\max }, k\right)} \prod_{i=w_{\min }}^{w_{\max }}\left(\binom{n_{i}}{a(i)} \cdot i^{2 a(i)}\right)\right) \tag{4.12}
\end{equation*}
$$

Consider the case where $w_{\min }=1$ and $w_{\max }=k$, and just take the one sequence $\mathbf{a}=$
$(1,1, \ldots, 1)$. Furthermore, because $n_{i}=\Theta\left(\frac{n}{i^{\beta}}\right)$ for every $i \in\left[w_{\min }, w_{\max }\right]$, let $\lambda$ be a universal constant such that $n_{i} \geq \frac{\lambda n}{i^{\beta}}$ for every $i$. Since $a(i)=1$ for all $i$ here, the quantity inside the $\Theta$ notation in (4.12), which we denote by $Q$, can be lower-bounded as follows

$$
\begin{aligned}
Q & \geq n \cdot\left(\frac{2^{k} k!}{m^{k}}\right) \cdot \prod_{i=w_{\min }}^{w_{\max }}\left(n_{i} \cdot i^{2}\right) \\
& \geq n \cdot\left(\frac{2^{k} k!}{m^{k}}\right) \cdot \prod_{i=1}^{k}\left(\left(\lambda n / i^{\beta}\right) \cdot i^{2}\right) \\
& =n \cdot\left(\frac{2^{k} \lambda^{k} n^{k} k!}{m^{k}}\right) \cdot \prod_{i=1}^{k} i^{-c_{0}} \\
& =n \cdot\left(\frac{2^{k} \lambda^{k} n^{k}}{m^{k}}\right) \cdot(k!)^{1-c_{0}} \\
& =\operatorname{poly}\left(n, 2^{k}\right) \cdot(k!)^{1-c_{0}}
\end{aligned}
$$

Because $c_{0}<1$, we have that $(k!)^{1-c_{0}}$ grows faster than poly $\left(2^{k}\right)$. Hence, $Q=\omega\left(\operatorname{poly}\left(n, 2^{k}\right)\right)$ and consequently $\ell_{k}=\omega\left(\operatorname{poly}\left(n, 2^{k}\right)\right)$.

Using reasoning similar to that used in Corollary 4.3.4, we see that Lemma 4.3.5 implies that under this stochastic regime, our proof approach cannot provide meaningful results for MinInf-CL. Thus we see a phase transition for the expected number of paths of any length $k$ : from at least $(k!)^{\Omega(1)}$ to $\operatorname{poly}\left(n, 2^{k}\right)$ at $\beta=3$. It is an open question what happens when $\beta=3$.

### 4.4 A Deterministic Rounding SAA Approach

In this section we revisit the SAA approach of Section 4.2, and instead of a randomized rounding, we apply a simple deterministic rounding scheme. The advantage of the latter is that the success probability of the algorithm no longer relies on the value $\Gamma$. However, this comes at
the expense of much worse bicriteria factors.
Once again we are going to independently sample $N=\frac{3 n}{\epsilon^{2}} \log \left(n^{2} \cdot 2^{m+1}\right)$ graphs $G_{j}=$ $\left(V, E_{j}\right)$ from $G(\vec{p})$, and then construct LP (4.1)-(4.4). Let $(x, y)$ be the optimal fractional solution of the LP. In this case, our returned solution will be $F_{0}=\left\{e \in E: x_{e} \geq \frac{1}{4 n^{2 / 3}}\right\}$. Before we proceed with our analysis, let us recall some important notation from Section 4.2. For any fixed $F \subseteq E, h\left(G_{j}, F\right)=\inf \left(V, E_{j} \backslash F, s\right)$ and $h(G, F)=\frac{1}{N} \sum_{j=1}^{N} h\left(G_{j}, F\right)$. Finally, $F^{*}$ denotes the optimal edge set for the given instance of MinInFEdge, and $\hat{F}$ denotes the optimal integral solution of LP (4.1)-(4.4).

Theorem 4.4.1. Whp, $F_{0}$ is an $\left(O\left(n^{2 / 3}\right), O\left(n^{2 / 3}\right)\right)$-approximation for MinInfEdge.

Proof. To begin with, by the definition of $F_{0}$ and constraint (4.3), we have

$$
\sum_{e \in F_{0}} c_{e} \leq 4 n^{2 / 3} \sum_{e \in F_{0}} c_{e} x_{e} \leq 4 n^{2 / 3} B
$$

Moving forward, note that by Lemma 4.2.1, we have $h\left(G, F^{*}\right) \leq(1+\epsilon) \mathbb{E}\left[h\left(G, F^{*}\right)\right]$ and $h\left(G, F_{0}\right) \geq(1-\epsilon) \mathbb{E}\left[h\left(G, F_{0}\right)\right]$ with probability at least $1-O\left(1 / n^{2}\right)$. If we show that $h\left(G, F_{0}\right) \leq 2 n^{2 / 3} h(G, \hat{F})$, then we are done. This is because:

$$
\begin{aligned}
\mathbb{E}\left[h\left(G, F_{0}\right)\right] & \leq \frac{h\left(G, F_{0}\right)}{1-\epsilon} \\
& \leq \frac{2 n^{2 / 3}}{1-\epsilon} h(G, \hat{F}) \\
& \leq \frac{2 n^{2 / 3}}{1-\epsilon} h\left(G, F^{*}\right) \\
& \leq \frac{2(1+\epsilon) n^{2 / 3}}{1-\epsilon} \mathbb{E}\left[h\left(G, F^{*}\right)\right]
\end{aligned}
$$

At first, suppose $h(G, \hat{F})>n^{1 / 3}$. Since $h\left(G_{j}, F_{0}\right) \leq n \leq n^{2 / 3} h(G, \hat{F})$ for any $j$, $h\left(G, F_{0}\right) \leq 2 n^{2 / 3} h(G, \hat{F})$ follows trivially through the definition of $h\left(G, F_{0}\right)$.

Next, suppose $h(G, \hat{F}) \leq n^{1 / 3}$. This implies

$$
\frac{1}{N} \sum_{j \in[N]} \sum_{v \in V}\left(1-y_{v j}\right) \leq h(G, \hat{F}) \leq n^{1 / 3}
$$

because the optimal LP-value is a lower bound for $h(G, \hat{F})$. Let now $A^{\prime}=\left\{j \in[N]: \sum_{v \in V}(1-\right.$ $\left.\left.y_{j v}\right) \leq n^{2 / 3}\right\}$ and $A^{\prime \prime}=[N] \backslash A^{\prime}=\left\{j \in[N]: \sum_{v \in V}\left(1-y_{v j}\right)>n^{2 / 3}\right\}$. The upper bound of the optimal fractional solution value then gives $\left|A^{\prime \prime}\right| \leq N / n^{1 / 3}$. Consider now any $j \in A^{\prime}$, and let $v$ be a node such that $1-y_{v j} \leq 1 / 2$. We will argue below that for any path $P \in \mathcal{P}\left(s, v, G_{j}\right)$, there exists an edge $e \in P$ such that $e \in F_{0}$. This means that if $v$ is infected in $\left(V, E_{j} \backslash F_{0}\right)$, then $1-y_{v j}>1 / 2$, and so $h\left(G_{j}, F_{0}\right) \leq \sum_{v} 2\left(1-y_{v j}\right)$. Hence,

$$
\begin{aligned}
h\left(G, F_{0}\right) & =\frac{1}{N} \sum_{j \in A^{\prime}} h\left(G_{j}, F_{0}\right)+\frac{1}{N} \sum_{j \in A^{\prime \prime}} h\left(G_{j}, F_{0}\right) \\
& \leq \frac{1}{N} \sum_{j \in A^{\prime}} \sum_{v \in V} 2\left(1-y_{v j}\right)+\frac{n\left|A^{\prime \prime}\right|}{N} \\
& \leq \frac{1}{N} \sum_{j \in[N]} \sum_{v \in V} 2\left(1-y_{v j}\right)+n^{2 / 3} \\
& \leq \frac{1}{N} \sum_{j \in[N]} \sum_{v \in V} 2\left(1-y_{v j}\right)+n^{2 / 3} h(G, \hat{F}) \\
& \leq\left(2+n^{2 / 3}\right) h(G, \hat{F}) \\
& \leq 2 n^{2 / 3} h(G, \hat{F})
\end{aligned}
$$

where the third inequality follows because $h\left(G_{j}, \hat{F}\right) \geq 1$, and thus $h(G, \hat{F}) \geq 1$.

Finally, we prove that for any $j \in A^{\prime}$, and any $v$ such that $1-y_{v j} \leq 1 / 2$, it must be the case that for each $P \in \mathcal{P}\left(s, v, G_{j}\right)$ we have $P \cap F_{0} \neq \emptyset$. Let $P=\left(v_{0}, v_{1}, \ldots, v_{r}\right)$ with $v_{0}=s$ be such a path of $\mathcal{P}\left(s, v, G_{j}\right)$. First, suppose $|P|=r \leq 2 n^{2 / 3}$. Then, constraint (4.2) yields $\sum_{e \in P} x_{e} \geq 1 / 2$, and hence there exists $e \in P$ with $x_{e} \geq 1 /(2|P|) \geq 1 /\left(4 n^{2 / 3}\right)$, which implies $e \in F_{0}$. Next, suppose $|P|>2 n^{2 / 3}$. Let $P^{\prime}=\left(v_{0}, v_{1}, \ldots, v_{k}\right)$ be the prefix of $P$ of length $k=2 n^{2 / 3}$. We will show that $P^{\prime} \cap F_{0} \neq \emptyset$, which implies $P \cap F_{0} \neq \emptyset$. By definition of $A^{\prime}$, we have $\sum_{i=0}^{k}\left(1-y_{v_{i} j}\right) \leq n^{2 / 3}$. Since $k=2 n^{2 / 3}$, there exists some $1 \leq \ell \leq k$ such that $1-y_{v_{\ell} j} \leq 1 / 2$, or $y_{v_{\ell} j} \geq 1 / 2$. Constraint (4.2) applied for $v_{\ell}$ and the path $\left(v_{0}, v_{1}, \ldots, v_{\ell}\right)$ gives $\sum_{i=0}^{\ell-1} x_{\left(v_{i}, v_{i+1}\right)} \geq y_{v_{\ell} j} \geq 1 / 2$, and thus there exists an edge $\left(v_{i}, v_{i+1}\right) \in P^{\prime}$ with $x_{\left(v_{i}, v_{i+1}\right)} \geq \frac{1}{2 \ell} \geq$ $\frac{1}{2 k} \geq \frac{1}{4 n^{2 / 3}}$, which means that $\left(v_{i}, v_{i+1}\right) \in F_{0}$.

## Chapter 5: Addressing Two-Stage Stochastic Clustering

### 5.1 Notation and Important Subroutines

For $k \in \mathbb{N}$, we use $[k]$ to denote $\{1,2, \ldots, k\}$. Also, for a vector $\alpha=\left(\alpha_{1}, \alpha_{2}, \ldots, \alpha_{k}\right)$ and a subset $X \subseteq[k]$, we use $\alpha(X)$ to denote $\sum_{i \in X} \alpha_{i}$. For a client $j$ and $R \geq 0$, we define $G_{j, R}=\{i \in \mathcal{F}: d(i, j) \leq R\}$ as the ball of radius $R$ around $j, i_{j, R}^{I}=\arg \min _{i \in G_{j, R}} c_{i}^{I}$ as the cheapest stage-I facility in that ball, and $i_{j, R}^{A}=\arg \min _{i \in G_{j, R}} c_{i}^{A}$ as the cheapest stage-II facility under scenario $A$ in the ball.

We repeatedly use a key subroutine named GreedyCluster() and shown in Algorithm 9. Its input is a set of clients $\mathcal{Q}$, a target radius $R$, and an ordering function $g: \mathcal{Q} \mapsto \mathbb{R}$. Its output is a set $H \subseteq \mathcal{Q}$ along with a mapping $\pi: \mathcal{Q} \mapsto H$. The goal of this subroutine is to sparsify the given input $\mathcal{Q}$, by greedily choosing a set of representative clients $H$.

```
Algorithm 9: GreedyCluster \((\mathcal{Q}, R, g)\)
    \(H \leftarrow \emptyset ;\)
    for each \(j \in \mathcal{Q}\) in non-increasing order of \(g(j)\) do
        \(H \leftarrow H \cup\{j\} ;\)
        for each \(j^{\prime} \in \mathcal{Q}\) with \(G_{j, R} \cap G_{j^{\prime}, R} \neq \emptyset\) do
                \(\pi\left(j^{\prime}\right) \leftarrow j, \mathcal{Q} \leftarrow \mathcal{Q} \backslash\left\{j^{\prime}\right\} ;\)
        end
    end
    Return ( \(H, \pi\) ) ;
```

The next observation follows trivially from the way Algorithm 9 operates.

Observation 5.1.1. For the outcome $(H, \pi)=\operatorname{GreedyCluster}(\mathcal{Q}, R, g)$ of Algorithm 9, the following two properties hold:

1. For all $j, j^{\prime} \in H$ with $j \neq j^{\prime}$, we have $G_{j, R} \cap G_{j^{\prime}, R}=\emptyset$.
2. For all $j \in \mathcal{Q}$ with $j^{\prime}=\pi(j)$, we have $G_{j, R} \cap G_{j^{\prime}, R} \neq \emptyset, d\left(j, j^{\prime}\right) \leq 2 R$, and $g\left(j^{\prime}\right) \geq g(j)$.

### 5.2 Generalizing to the Black-Box Setting

Let $\mathcal{P}$ be any of the two-stage problems we consider, with polynomial-scenarios variant $\mathcal{P}$ Poly and black-box variant $\mathcal{P}$-BB. Moreover, suppose that we have an $\eta$-approximation algorithm $\operatorname{Alg} \mathcal{P}$ for $\mathcal{P}-P o l y$, which we intend to use to solve $\mathcal{P}-\mathbf{B B}$. Before we proceed to our generalization scheme, we present some important definitions and assumptions.

As a starting point, assume that we are given a fixed radius demand $R$; we later discuss how to optimize over this. Hence, we will be denoting a $\mathcal{P}$ - $\mathbf{B B}$ problem instance by the tuple $\mathfrak{I}=\left(\mathcal{C}, \mathcal{F}, \mathcal{M}_{I}, c^{I}, B, R\right)$, where $\mathcal{C}$ is the set of clients, $\mathcal{F}$ the set of facilities $i$, each with stage-I $\operatorname{cost} c_{i}^{I}, \mathcal{M}_{I} \subseteq 2^{\mathcal{F}}$ the set of legal stage-I openings (representing the stage-I specific constraints of $\mathcal{P}$ ), $B$ the budget, and $R$ the given covering demand. In addition, there is anderlying distribution $\mathcal{D}$, where each scenario $A \in \mathcal{D}$ appears with some unknown probability $p_{A}$. Our only means of access to $\mathcal{D}$ is via a sampling oracle. Finally, when a scenario $A \in \mathcal{D}$ is revealed, we also learn the corresponding facility costs $c_{i}^{A}$.

Definition 5.2.1. We define a strategy $s$ to be a tuple $\left(F_{I}^{s}, F_{A}^{s} \mid A \in \mathcal{D}\right)$ of facility sets, where $A$ ranges over $\mathcal{D}$. The set $F_{I}^{s}$ represents the facilities the strategy $s$ opens in stage-I, and $F_{A}^{s}$ denotes the facilities $s$ opens in stage-II, when the arriving scenario is $A$. In other words, a strategy is a just potential (not necessarily feasible) solution for $\mathcal{P}-\mathbf{B B}$.

Assumption 5.2.2. For any strategy $s$ and $A \in \mathcal{D}$, the value $c^{A}\left(F_{A}^{s}\right)$ has a continuous CDF. We can assume this w.l.o.g.; we simply add a dummy facility $i_{d}$ in the input, and for all $s$ and $A \in \mathcal{D}$, we include $i_{d}$ in the original $F_{A}^{s}$. Then, $c_{i_{d}}^{A}$ is set to be some infinitesimal smooth noise. Also, $B$ and $\mathcal{M}_{I}$ can trivially be extended to account for $i_{d}$. Finally, the assumption implies that for any finite set of scenarios $Q$, the values $c^{A}\left(F_{A}^{s}\right)$ for all $A \in Q$ are distinct with probability 1.

We say that a given instance $\mathfrak{I}$ is feasible for $\mathcal{P}-\mathbf{B B}$, if there exists a strategy $s^{*}$ satisfying:

$$
F_{I}^{s^{*}} \in \mathcal{M}_{I}, \quad c^{I}\left(F_{I}^{s^{*}}\right)+\sum_{A \in \mathcal{D}} p_{A} c^{A}\left(F_{A}^{s^{*}}\right) \leq B, \quad \forall j \in A \in \mathcal{D} \quad d\left(j, F_{I}^{s^{*}} \cup F_{A}^{s^{*}}\right) \leq R
$$

For $\mathcal{P}$-Poly, consider an instance $\mathfrak{J}=\left(\mathcal{C}, \mathcal{F}, \mathcal{M}_{I}, Q, \vec{q}, \vec{c}, B, R\right)$, where $\mathcal{C}, \mathcal{F}, \mathcal{M}_{I}, B, R$ are as in the $\mathcal{P}$-BB setting, $Q$ is the set of provided scenarios, $\vec{c}$ the vector of stage-I and stage-II explicitly given costs, and $\vec{q}$ the vector of occurrence probabilities $q_{A}$ of each $A \in Q$. We say that the instance $\mathfrak{J}$ is feasible for $\mathcal{P}$-Poly, if there exist sets $F_{I} \subseteq \mathcal{F}$ and $F_{A} \subseteq \mathcal{F}$ for every $A \in Q$ :

$$
F_{I} \in \mathcal{M}_{I}, \quad c^{I}\left(F_{I}\right)+\sum_{A \in Q} q_{A} c^{A}\left(F_{A}\right) \leq B, \quad \forall j \in A \in Q \quad d\left(j, F_{I} \cup F_{A}\right) \leq R
$$

We also write $F$ for the overall collection of sets $F_{I}$ and $F_{A}: A \in Q$.

Definition 5.2.3. An algorithm $\operatorname{Alg} \mathcal{P}$ is a valid $\eta$-approximation algorithm for $\mathcal{P}$-Poly, if given any problem instance $\mathfrak{J}=\left(\mathcal{C}, \mathcal{F}, \mathcal{M}_{I}, Q, \vec{q}, \vec{c}, B, R\right)$, one of the following holds:

A1 If $\mathfrak{J}$ is feasible for $\mathcal{P}-\mathbf{P o l y}$, then $\operatorname{Alg} \mathcal{P}$ returns a collection of sets $F$ with $F_{I} \in \mathcal{M}_{I}$,

$$
c^{I}\left(F_{I}\right)+\sum_{A \in Q} q_{A} c^{A}\left(F_{A}\right) \leq B \text { and } \forall j \in A \in Q \quad d\left(j, F_{I} \cup F_{A}\right) \leq \eta R .
$$

A2 If $\mathfrak{J}$ is not feasible for $\mathcal{P}$-Poly, then the algorithm either returns "INFEASIBLE", or returns a collection of sets $F$ satisfying the properties presented in A1.

Definition 5.2.4. A valid $\eta$-approximation algorithm $\operatorname{Alg} \mathcal{P}$ for $\mathcal{P}$-Poly is efficiently generalizable, if for every instance $\mathfrak{J}=\left(\mathcal{C}, \mathcal{F}, \mathcal{M}_{I}, Q, \vec{q}, \vec{c}, B, R\right)$ for which it returns a solution $F$, there is an efficient procedure that implicitly extends this to a strategy $\bar{s}$, and satisfies:

S1 Given any $A \in \mathcal{D}$, it returns a set $F_{A}^{\bar{s}} \subseteq \mathcal{F}$, with $d\left(j, F_{I}^{\bar{s}} \cup F_{A}^{\bar{s}}\right) \leq \eta R$ for all $j \in A$.

S2 $F_{I}^{\bar{s}}=F_{I}$ and $F_{A}^{\bar{s}}=F_{A}$ for every $A \in Q$.

S3 Given $\mathfrak{J}$, let $\mathcal{S}$ be the set of all possible strategies that are potentially achievable using the extension procedure for any set $Q$. Then $|\mathcal{S}| \leq t_{\mathcal{P}}(n, m)$ for some function $t_{\mathcal{P}}(n, m)$, with $\log \left(t_{\mathcal{P}}(n, m)\right)=\operatorname{poly}(n, m)$.

Note that property $\mathbf{S 3}$ is not trivial, since by default $|\mathcal{S}| \leq 2^{m|\mathcal{D}|}$, and $|\mathcal{D}|$ can be exponentially large or even uncountably infinite.

The first step of our generalization is based on sampling a set $Q$ of scenarios from $\mathcal{D}$, and then applying the efficiently-generalizable $\operatorname{Alg} \mathcal{P}$ on $Q$. When running the latter, we also increase the available budget to $(1+\epsilon) B$, for some $\epsilon>0$. The purpose of this step is to verify whether or not the given instance of $\mathcal{P}-\mathbf{B B}$ is feasible, and to achieve this we may have to repeat the step a polynomial number of times. See Algorithm 10 for the full details.

If Algorithm 10 returns "INFEASIBLE", then our approach would deem that $\mathfrak{I}$ is not feasible for $\mathcal{P}$-BB. Otherwise, let $F$ be the solution returned by $\operatorname{Alg} \mathcal{P}$ at the last "successfull" iteration of the while loop. Because $\operatorname{Alg} \mathcal{P}$ is efficiently-generalizable, we can apply its extension procedure to any arriving scenario, and therefore implicitly construct a strategy $\bar{s}$. By the properties of $\operatorname{Alg} \mathcal{P}$ and $\mathbf{S 2}$, S1, we have $F_{I}^{\bar{s}} \in \mathcal{M}_{I}$ and $d\left(j, F_{I}^{\bar{s}} \cup F_{A}^{\bar{s}}\right) \leq \eta R$ for every $A \in \mathcal{D}$ and $j \in A$.

However, we are not yet done. The second step of our generalization framework consists of

```
Algorithm 10: Determining Feasibility for \(\mathcal{P}\)-BB.
    Input: Parameters \(\epsilon, \gamma, \alpha \in(0,1), N \geq 1\) and a \(\mathcal{P}\)-BB instance
            \(\mathfrak{I}=\left(\mathcal{C}, \mathcal{F}, \mathcal{M}_{I}, c^{I}, B, R\right)\).
    If \(\exists j \in \mathcal{C}: d(j, \mathcal{F})>R\) then return "INFEASIBLE"; // Points not sampled
    for \(h=1, \ldots,\left[\log _{\frac{13}{12}}(1 / \gamma)\right]\) do
        Draw \(N\) independent samples from the oracle, obtaining set \(Q=\left\{S_{1}, \ldots, S_{N}\right\}\);
        Let \(\vec{c}\) the vector containing \(c^{I}\) and the stage-II facility-cost vectors of all \(S_{v} \in Q\);
        For every \(S_{v} \in Q\) set \(q_{S_{v}} \leftarrow 1 / N\);
        if \(\operatorname{Alg} \mathcal{P}\left(\mathcal{C}, \mathcal{F}, \mathcal{M}_{I}, Q, \vec{q}, \vec{c},(1+\epsilon) B, R\right)\) returns \(F\) then
            Let \(T\) be the \(\lceil\alpha N\rceil^{\text {th }}\) largest value of \(c^{S_{v}}\left(F_{S_{v}}\right)\) among all scenarios in \(Q\);
            Return \((F, T)\);
        end
    end
    Return "INFEASIBLE";
```

slightly modifying the strategy $\bar{s}$. For that reason, we use the value $T$ returned by Algorithm 10, which corresponds to the $\lceil\alpha N\rceil^{\text {th }}$ largest value $c^{S_{v}}\left(F_{S_{v}}^{\bar{s}}\right)$ among all $S_{v} \in Q$, with $Q$ the sampled set in the last iteration of the while loop ( $F_{S_{v}}^{\overline{5}}=F_{S_{v}}$ by S2). Note here that Assumption 5.2.2 ensures that the choice of $T$ is well-defined.

If now an arriving scenario $A$ has $c^{A}\left(F_{A}^{\bar{S}}\right)>T$, we will perform no stage-II opening. This modification eventually constructs a new strategy $\hat{s}$, with $F_{I}^{\hat{s}}=F_{I}^{\bar{s}}, F_{A}^{\hat{s}}=\emptyset$ when $c^{A}\left(F_{A}^{\bar{s}}\right)>T$, and $F_{A}^{\hat{s}}=F_{A}^{\bar{s}}$ if $c^{A}\left(F_{A}^{\bar{s}}\right) \leq T$. The latter strategy will determine our final opening actions, and hence we need to analyze its opening cost $C(\hat{s})$ over $\mathcal{D}$, and the probability with which it does not return an $\eta$-approximate solution. Regarding the latter, note that when $F_{A}^{\hat{s}} \neq F_{A}^{\bar{s}}$, we can no longer guarantee an approximation ratio of $\eta$ as implied by property $\mathbf{S 1}$ for $\bar{s}$.

The next concentration lemma is crucial in our calculations, and it can be seen as an improved variant of Markov's inequality.

Lemma 5.2.5. [127] Let $X_{1}, \ldots, X_{K}$ be non-negative independent random variables, with expectations $\mu_{1}, \ldots, \mu_{K}$, where $\mu_{k} \leq 1$ for all $k$. Let $X=\sum_{k=1}^{K} X_{i}$, and let $\mu=\sum_{k=1}^{K} \mu_{i}=\mathbb{E}[X]$.

Then for all $\delta>0$ we have $\operatorname{Pr}[X<\mu+\delta] \geq \min \left\{\frac{\delta}{1+\delta}, \frac{1}{13}\right\}$.

Lemma 5.2.6. If instance $\mathfrak{I}$ is feasible for $\mathcal{P}-\boldsymbol{B B}$ and $N \geq 1 / \epsilon$, then with probability at least $1-\gamma$ Algorithm 10 does not terminate with "INFEASIBLE".

Proof. By rescaling, we assume w.l.o.g. that $B=1$. Also, observe that the cost of any strategy $s$ over $\mathcal{D}$ can be expressed as

$$
C(s)=c^{I}\left(F_{I}^{s}\right)+\sum_{A \in \mathcal{D}} p_{A} c^{A}\left(F_{A}^{s}\right)
$$

For any specific execution of the while loop in Algorithm 10, let $Y_{v}^{s}$ be the second-stage cost of $s$ on sample $S_{v}$. Finally, for a fixed $s$ the random variables $Y_{v}^{s}$ are independent, and the empirical cost of $s$ on $Q$ can be expressed as

$$
\hat{C}(s)=c^{I}\left(F_{I}^{s}\right)+\frac{1}{N} \sum_{v=1}^{N} Y_{v}^{s}
$$

If $\mathfrak{I}$ is feasible, then there exists strategy $s^{\star}$ with $F_{I}^{s^{\star}} \in \mathcal{M}_{I}$ and $d\left(j, F_{I}^{s^{\star}} \cup F_{A}^{s^{\star}}\right) \leq R$ for all $A \in Q$ and $j \in A$. We will also show that $\hat{C}\left(s^{\star}\right) \leq(1+\epsilon) B$ with probability at least $1 / 13$. In this case, the restriction of $s^{\star}$ to $Q$ verifies that $\left(\mathcal{C}, \mathcal{F}, \mathcal{M}_{I}, Q, \vec{q}, \vec{c},(1+\epsilon) B, R\right)$ is feasible for $\mathcal{P}$-Poly. Thus, since $\operatorname{Alg} \mathcal{P}$ is a valid $\eta$-approximation for $\mathcal{P}$-Poly, it will not return "INFEASIBLE".

As $s^{*}$ is feasible for $\mathfrak{I}$ we have $C\left(s^{\star}\right) \leq B$, implying $\mathbb{E}\left[Y_{v}^{s^{\star}}\right]=\sum_{A \in \mathcal{D}} p_{A} \cdot c^{A}\left(F_{A}^{s^{\star}}\right) \leq B=$ 1 for all samples $v$. By Lemma 5.2.5 with $\delta=\epsilon B N$, this yields

$$
\operatorname{Pr}\left[\sum_{v=1}^{N} Y_{v}^{s^{\star}}<\mathbb{E}\left[\sum_{v=1}^{N} Y_{v}^{s^{\star}}\right]+\epsilon B N\right] \geq \min \left\{\frac{\epsilon B N}{1+\epsilon B N}, \frac{1}{13}\right\}
$$

When $N \geq \frac{B}{\epsilon}=\frac{1}{\epsilon}$, we see that $\epsilon B N /(1+\epsilon B N) \geq 1 / 13$. Hence, with probability at least $1 / 13$ we have $\sum_{v=1}^{N} Y_{v}^{s^{\star}}<\mathbb{E}\left[\sum_{v=1}^{N} Y_{v}^{s^{\star}}\right]+\epsilon B N$, in which case $\hat{C}\left(s^{*}\right) \leq(1+\epsilon) B$ as shown below:

$$
\begin{aligned}
\hat{C}\left(s^{\star}\right) & =c^{I}\left(F_{I}^{s^{\star}}\right)+\frac{1}{N} \sum_{v=1}^{N} Y_{v}^{s^{\star}} \\
& \leq c^{I}\left(F_{I}^{s^{\star}}\right)+\frac{1}{N} \sum_{v=1}^{N} \mathbb{E}\left[Y_{v}^{s^{\star}}\right]+\epsilon B \\
& \leq c^{I}\left(F_{I}^{s^{\star}}\right)+\sum_{A \in \mathcal{D}} p_{A} \cdot c^{A}\left(F_{A}^{s^{\star}}\right)+\epsilon B \\
& \leq(1+\epsilon) B
\end{aligned}
$$

So each iteration terminates successfully with probability at least $1 / 13$. To bring the error probability down to at most $\gamma$, we repeat the process for $\left[\log _{\frac{13}{12}}(1 / \gamma)\right\rceil$ iterations.

Let $\mathcal{T}$ be the event that Algorithm 10 terminates without returning "INFEASIBLE", and $\mathcal{T}_{h}$ the event that $\operatorname{Alg} \mathcal{P}$ found a solution $F$ at the $h^{\text {th }}$ iteration of the while loop. We denote by Invalid the event that Algorithm 10 returns an invalid output; specifically, if $\mathcal{T}$ occurs, Invalid is the event of having $C(\hat{s})>(1+2 \epsilon) B$, otherwise it is the event of mistakenly deciding that $\mathfrak{I}$ is not feasible. Let now $Q_{h}$ be the set of scenarios sampled at the $h^{\text {th }}$ iteration of Algorithm 10, and for any strategy $s$ let $T_{s}^{h}$ be the $\lceil\alpha N\rceil^{\text {th }}$ largest value $c^{S_{v}}\left(F_{S_{v}}^{s}\right)$ among all $S_{v} \in Q_{h}$. We then denote by $\mathcal{E}_{h}$ the event that for all $s \in \mathcal{S}$, we have $\operatorname{Pr}_{A \sim \mathcal{D}}\left[c^{A}\left(F_{A}^{s}\right)>T_{s}^{h}\right] \geq \frac{\alpha}{4}$. Finally, note that due to $\mathbf{S} \mathbf{3}$ the set $\mathcal{S}$ is deterministically given in the event $\mathcal{E}_{h}$.

The two following lemmas are standard Chernoff bounds.

Lemma 5.2.7. Let $X_{1}, X_{2}, \ldots, X_{K}$ be independent random variables with $X_{k} \in[0,1]$ for all $k$. For $X=\sum_{k=1}^{K} X_{k}$ with $\mu=\mathbb{E}[X]$ and any $\delta>0$, we have $\operatorname{Pr}[X \leq(1-\delta) \mu] \leq e^{\frac{-\mu \delta^{2}}{2}}$.

Lemma 5.2.8. Let $X_{1}, X_{2}, \ldots, X_{K}$ be independent Bernoulli random variables with parameter p. Let $X=\sum_{k=1}^{K} X_{k}$ the corresponding binomial random variable. If for the realization of $X$ we have $X=q K$, then for any $\delta>0$ we have $\operatorname{Pr}[p<q-\delta] \leq e^{-K \delta^{2} / 2 p}$

Lemma 5.2.9. For $\gamma, \alpha \in(0,1)$ and $N=\mathcal{O}\left(\frac{1}{\alpha} \log \left(\frac{t_{\mathcal{P}}(n, m)}{\gamma}\right)\right), \operatorname{Pr}\left[\overline{\mathcal{E}}_{h}\right] \leq \gamma /\left(\log _{\frac{13}{12}}\left(\frac{1}{\gamma}\right)+1\right)$.

Proof. Focus on a specific iteration $h$. Consider a strategy $s \in \mathcal{S}$, and for each $S_{v} \in Q_{h}$ let $X_{v}$ be an indicator random variable that is $1 \operatorname{iff} C^{S_{v}}\left(F_{S_{v}}^{s}\right)>T_{s}^{h}$. Also let $X=\sum_{v=1}^{N} X_{v}$, and note that by Assumption 5.2.2 we have $X=\lceil\alpha N\rceil-1$. This implies that the empirical probability of scenarios with stage-II cost more than $T_{s}^{h}$ is $q_{s}^{h}=(\lceil\alpha N\rceil-1) / N$. Finally, let $p_{s}^{h}=\operatorname{Pr}_{A \sim \mathcal{D}}\left[c^{A}\left(F_{A}^{s}\right)>T_{s}^{h}\right]$. If $p_{s}^{h} \geq \alpha$ then we immediately get $\operatorname{Pr}\left[p_{s}^{h}<\alpha / 4\right]=0$. Therefore, assume that $p_{s}^{h}<\alpha$. If $N \geq 4 / \alpha$ then we have:

$$
q_{s}^{h}-\frac{\alpha}{2}=\frac{\lceil\alpha N\rceil-1}{N}-\frac{\alpha}{2} \geq \frac{\alpha N-1}{N}-\frac{\alpha}{2}=\frac{\alpha}{2}-\frac{1}{N} \geq \frac{\alpha}{2}-\frac{\alpha}{4}=\frac{\alpha}{4}
$$

Hence, if $p_{s}^{h} \geq q_{s}^{h}-\frac{\alpha}{2}$ and $N \geq 4 / \alpha$, we get $p_{s}^{h} \geq \frac{\alpha}{4}$. Using Lemma 5.2.8 with $p_{s}^{h}<\alpha, \delta=\alpha / 2$ and $N=\frac{8}{\alpha} \log \left(\frac{t_{\mathcal{P}}(n, m)}{\gamma}\left(\log _{\frac{13}{12}}\left(\frac{1}{\gamma}\right)+1\right)\right) \geq 4 / \alpha$ yields the following:

$$
\begin{aligned}
\operatorname{Pr}\left[p_{s}^{h}<\frac{\alpha}{4}\right] & \leq \operatorname{Pr}\left[p_{s}^{h}<q_{s}^{h}-\alpha / 2\right] \\
& \leq e^{-\left(\frac{t_{\mathcal{p}(n, m)}^{\gamma}\left(\log _{\frac{13}{12}}\left(\frac{1}{\gamma}\right)+1\right)}{}\right)} \\
& =\frac{\gamma}{t_{\mathcal{P}}(n, m)\left(\log _{\frac{13}{12}}(1 / \gamma)+1\right)}
\end{aligned}
$$

A union bound over all $s \in \mathcal{S}$ and property $\mathbf{S 3}$ will finally give $\operatorname{Pr}\left[\overline{\mathcal{E}}_{h}\right] \leq \gamma /\left(\log _{\frac{13}{12}}\left(\frac{1}{\gamma}\right)+1\right)$.

Theorem 5.2.10. For any $\epsilon, \gamma, \alpha \in(0,1)$ and $N=\mathcal{O}\left(\frac{1}{\epsilon \alpha} \log \left(\frac{t_{\boldsymbol{\mathcal { P }}}(n, m)}{\gamma}\right)\right)$, we get $\operatorname{Pr}[$ Invalid $] \leq 3 \gamma$.

Proof. Using the definition of the Invalid event and Lemmas 5.2.6, 5.2.9 we get the following.

$$
\begin{align*}
\operatorname{Pr}[\text { Invalid }] & =\operatorname{Pr}[\text { Invalid } \mid \overline{\mathcal{T}}] \operatorname{Pr}[\overline{\mathcal{T}}]+\operatorname{Pr}[\text { Invalid } \mid \mathcal{T}] \operatorname{Pr}[\mathcal{T}] \leq \gamma+\sum_{h} \operatorname{Pr}\left[\operatorname{Invalid} \wedge \mathcal{T}_{h}\right] \\
& =\gamma+\sum_{h}\left(\operatorname{Pr}\left[\operatorname{Invalid} \wedge \mathcal{T}_{h} \mid \mathcal{E}_{h}\right] \operatorname{Pr}\left[\mathcal{E}_{h}\right]+\operatorname{Pr}\left[\operatorname{Invalid} \wedge \mathcal{T}_{h} \mid \overline{\mathcal{E}}_{h}\right] \operatorname{Pr}\left[\overline{\mathcal{E}}_{h}\right]\right) \\
& \leq 2 \gamma+\sum_{h} \operatorname{Pr}\left[\operatorname{Invalid} \wedge \mathcal{T}_{h} \wedge \mathcal{E}_{h}\right] \tag{5.1}
\end{align*}
$$

For each $s \in \mathcal{S}$, let $t_{s}$ be value such that $\operatorname{Pr}_{A \sim \mathcal{D}}\left[c^{A}\left(F_{A}^{s}\right)>t_{s}\right]=\frac{\alpha}{4}$. Note that the existence of $t_{s}$ is guaranteed by Assumption 5.2.2. Further, for each $s \in \mathcal{S}, A \in \mathcal{D}$, define $\tilde{c}^{A}\left(F_{A}^{s}\right)$ to be $c^{A}\left(F_{A}^{s}\right)$ if $c^{A}\left(F_{A}^{s}\right) \leq t_{s}$, and 0 otherwise. In addition, for an iteration $h$ let $Y_{v, h}^{s}$ be a random variable denoting the second-stage $\tilde{c}$ cost of $s$ for the $v$-th sample of $h$, and $Z_{v, h}^{s}$ be an indicator random variable that is 1 iff the original second-stage cost of $s$ on the $v$-th sample of $h$ is greater than $t_{s}$. We use the following cost functions:

$$
\hat{C}_{h}(s)=c^{I}\left(F_{I}^{s}\right)+\frac{1}{N} \sum_{v=1}^{N} Y_{v, h}^{s}+\frac{t_{s}}{N} \sum_{v=1}^{N} Z_{v, h}^{s} \text { and } \tilde{C}(s)=c^{I}\left(F_{I}^{s}\right)+\sum_{A \in \mathcal{D}} p_{A} \cdot \tilde{c}^{A}\left(F_{A}^{s}\right)
$$

Also, if $p_{s}=\operatorname{Pr}_{A \sim \mathcal{D}}\left[c^{A}\left(F_{A}^{s}\right)>t_{s}\right]$, then $\mathbb{E}\left[\hat{C}_{h}(s)\right]=\tilde{C}(s)+p_{s} t_{s}$. Finally, let $\hat{C}_{h}^{I I}(s)=\hat{C}_{h}(s)-$ $c^{I}\left(F_{I}^{s}\right)$ and $\tilde{C}^{I I}(s)=\tilde{C}(s)-c^{I}\left(F_{I}^{s}\right)$.

Now observe that if Invalid $\wedge \mathcal{T}_{h} \wedge \mathcal{E}_{h}$ occurs, then there must exist some $s \in \mathcal{S}$ with $\hat{C}_{h}(s) \leq$ $(1+\epsilon) B$ and $\tilde{C}(s)>(1+2 \epsilon) B$. Specifically we have $\hat{C}_{h}(\bar{s}) \leq(1+\epsilon) B$ and $\tilde{C}(\bar{s})>(1+2 \epsilon) B$. To see why $\hat{C}_{h}(\bar{s}) \leq(1+\epsilon) B$ is true, note than under this event $\operatorname{Alg} \mathcal{P}$ finds a solution in iteration $h$. The empirical cost of this solution (which corresponds to a restriction of $\bar{s}$ ) is at most $(1+\epsilon) B$, and the pruning based on the value $t_{s}$ can only decrease this cost. Regarding $\tilde{C}(\bar{s})>(1+2 \epsilon) B$,
under Invalid $\wedge \mathcal{T}_{h} \wedge \mathcal{E}_{h}$ we at first have $C(\hat{s})>(1+2 \epsilon) B$. In addition, $\tilde{C}(\bar{s}) \leq C(\hat{s})$, because by the definitions of $t_{s}$ and $\mathcal{E}_{h}$ we have $t_{s} \geq T_{s}^{h}$. Hence, we upper bound the probability of Invalid $\wedge \mathcal{T}_{h} \wedge \mathcal{E}_{h}$ as follows:

$$
\begin{align*}
\operatorname{Pr}\left[\operatorname{Invalid} \wedge \mathcal{T}_{h} \wedge \mathcal{E}_{h}\right] & \leq \operatorname{Pr}\left[\exists s \in \mathcal{S}: \hat{C}_{h}(s) \leq(1+\epsilon) B \wedge \tilde{C}(s)>(1+2 \epsilon) B\right] \\
& \leq \operatorname{Pr}\left[\exists s \in \mathcal{S}: \hat{C}_{h}(s) \leq(1+\epsilon) B \wedge \tilde{C}(s)+p_{s} t_{s}>(1+2 \epsilon) B+p_{s} t_{s}\right] \\
& \leq \operatorname{Pr}\left[\exists s \in \mathcal{S}: \hat{C}_{h}(s) \leq(1+\epsilon) B \wedge \mathbb{E}\left[\hat{C}_{h}(s)\right]>(1+2 \epsilon) B+p_{s} t_{s}\right] \\
& \leq \operatorname{Pr}\left[\exists s \in \mathcal{S}: \hat{C}_{h}^{I I}(s) \leq\left(1-\delta_{s}\right) \mathbb{E}\left[\hat{C}_{h}^{I I}(s)\right]\right] \\
& \leq \sum_{s \in \mathcal{S}} \operatorname{Pr}\left[\hat{C}_{h}^{I I}(s) \leq\left(1-\delta_{s}\right) \mathbb{E}\left[\hat{C}_{h}^{I I}(s)\right]\right] \\
& =\sum_{s \in \mathcal{S}} \operatorname{Pr}\left[N \cdot \hat{C}_{h}^{I I}(s) / t_{s} \leq\left(1-\delta_{s}\right) N \cdot \mathbb{E}\left[\hat{C}_{h}^{I I}(s)\right] / t_{s}\right] \tag{5.2}
\end{align*}
$$

In the above we defined $\delta_{s}$ such that $\delta_{s} \geq \frac{\epsilon+p_{s} t_{s}}{1+2 \epsilon+p_{s} t_{s}}$, and also we made use of $B=1$ and $\mathbb{E}\left[\hat{C}_{h}(s)\right]=\tilde{C}(s)+p_{s} t_{s}>1+2 \epsilon+p_{s} t_{s}$. Applying Lemma 5.2 .7 gives

$$
\begin{equation*}
\operatorname{Pr}\left[N \cdot \hat{C}_{h}^{I I}(s) / t_{s} \leq\left(1-\delta_{s}\right) N \cdot \mathbb{E}\left[\hat{C}_{h}^{I I}(s)\right] / t_{s}\right] \leq e^{\frac{-N\left(\epsilon+p_{s} t_{s}\right)^{2}}{2 t_{s}\left(1+2 \epsilon+p_{s} t_{s}\right)}} \tag{5.3}
\end{equation*}
$$

We now focus on the quantity $\frac{\left(\epsilon+p_{s} t_{s}\right)^{2}}{2 t_{s}\left(1+2 \epsilon+p_{s} t_{s}\right)}$, and consider two distinct cases for $p_{s} t_{s}$.

- Suppose $p_{s} t_{s} \geq \epsilon$. Then:

$$
\frac{\left(\epsilon+p_{s} t_{s}\right)^{2}}{2 t_{s}\left(1+2 \epsilon+p_{s} t_{s}\right)} \geq \frac{p_{s}^{2} t_{s}^{2}}{2 t_{s}\left(1+3 p_{s} t_{s}\right)} \geq \frac{p_{s}}{2} \frac{p_{s} t_{s}}{1+3 p_{s} t_{s}} \geq \frac{\epsilon \cdot p_{s}}{2(1+3 \epsilon)}
$$

where the last inequality follows because $x /(1+3 x)$ is increasing and in our case $x \geq \epsilon$.

- Suppose $p_{s} t_{s}<\epsilon$. Then:

$$
\frac{\left(\epsilon+p_{s} t_{s}\right)^{2}}{2 t_{s}\left(1+2 \epsilon+p_{s} t_{s}\right)} \geq \frac{\epsilon^{2}}{2 t_{s}(1+3 \epsilon)} \geq \frac{\epsilon \cdot p_{s}}{2(1+3 \epsilon)}
$$

where in the last inequality we used the fact that in this case $t_{s}<\epsilon / p_{s}$.

Therefore, by definition of $p_{s}$, we have $\frac{\left(\epsilon+p_{s} t_{s}\right)^{2}}{2 t_{s}\left(1+2 \epsilon+p_{s} t_{s}\right)} \geq \frac{\epsilon \cdot \alpha}{8(1+3 \epsilon)}$ in every case. Plugging that in (5.3), (5.2), and setting $N=\frac{8(1+\epsilon)}{\epsilon \alpha} \log \left(\frac{t_{p}(n, m)}{\gamma}\left(\log _{\frac{13}{12}}\left(\frac{1}{\gamma}\right)+1\right)\right)$ gives $\operatorname{Pr}\left[\operatorname{Invalid} \wedge \mathcal{T}_{h} \wedge \mathcal{E}_{h}\right] \leq$ $\gamma /\left(\log _{\frac{13}{12}}\left(\frac{1}{\gamma}\right)+1\right)$. Finally, using this in (5.1) gives an error probability of at most $3 \gamma$.

Theorem 5.2.11. For any $\gamma, \alpha \in(0,1)$ and $N=\mathcal{O}\left(\frac{1}{\alpha} \log \left(\frac{t_{p}(n, m)}{\gamma}\right)\right)$, the solution strategy $\hat{s}$ satisfies $\operatorname{Pr}_{A \sim \mathcal{D}}\left[d\left(j, F_{I}^{\hat{s}} \cup F_{A}^{\hat{s}}\right) \leq \eta R, \forall j \in A\right] \geq 1-2 \alpha$ with probability at least $1-\gamma$.

Proof. Consider some iteration $h$ and strategy $s \in \mathcal{S}$. Let

$$
p_{s}^{h}=\operatorname{Pr}_{A \sim \mathcal{D}}\left[c^{A}\left(F_{A}^{s}\right)>T_{s}^{h}\right]
$$

We define $\mathcal{B}_{s}^{h}$ to be the event of having $p_{T_{s}^{h}}>2 \alpha$. Suppose that $p_{s}^{h}>\alpha$, otherwise $\mathcal{B}_{s}^{h}$ cannot occur. Let $X_{v}$ an indicator random variable that is 1 iff $s$ has stage-II cost larger than $T_{s}^{h}$ in the $v$-th sample. Also, let $X=\sum_{v=1}^{N} X_{v}$, and recall that $X=\lceil\alpha N\rceil-1 \leq \alpha N$. Moreover, we have $\mathbb{E}[X]=p_{s}^{h} N$ and notice that $2 X>\mathbb{E}[X]$ implies $p_{s}^{h}<2 \alpha$. Using Lemma 5.2 .7 with $\delta=1 / 2$ we get $\operatorname{Pr}[X \leq \mathbb{E}[X] / 2] \leq e^{-p_{s}^{h} N / 8}$. Because $p_{s}^{h}>\alpha$, setting

$$
N=\frac{8}{\alpha} \log \left(\frac{t_{\mathcal{P}}(n, m)}{\gamma}\left(\log _{\frac{13}{12}}\left(\frac{1}{\gamma}\right)+1\right)\right)
$$

and using a union bound gives $\sum_{h} \sum_{s \in \mathcal{S}} \operatorname{Pr}\left[\mathcal{B}_{s}^{h}\right] \leq \gamma$.

Finally, by optimizing over the radius, we get our main generalization result that does not only work for a fix radius guess $R$.

Theorem 5.2.12. Assume we have an efficiently generalizable $\eta$-approximation algorithm for $\mathcal{P}$-Poly. Then, using $\mathcal{O}\left(\frac{1}{\epsilon \alpha} \log \left(\frac{n m \cdot t_{p}(n, m)}{\gamma}\right) \log \frac{n m}{\gamma}\right)$ samples, we obtain a strategy $\hat{s}$ and a radius $R$, such that with probability at least $1-\mathcal{O}(\gamma)$ the following hold: (i) $C(\hat{s}) \leq(1+2 \epsilon) B$, (ii) $F_{I}^{\hat{s}} \in \mathcal{M}_{I}$; (iii) $R \leq R^{*}$, where $R^{*}$ is the optimal radius for the given $\mathcal{P}$-BB instance; (iv) $\operatorname{Pr}_{A \sim \mathcal{D}}\left[d\left(j, F_{I}^{\hat{s}} \cup F_{A}^{\hat{s}}\right) \leq \eta R, \forall j \in A\right] \geq 1-2 \alpha$.

Proof. Because $R^{*}$ is the distance between some facility and some client, there are at most $n m$ alternatives for it. Thus, we can run Algorithm 10 for all possible $n m$ target radius values, using error parameter $\gamma^{\prime}=\frac{\gamma}{n m}$. We then return the smallest radius that did not yield "INFEASIBLE". By a union bound over all radius choices, the probability of the Invalid event in any of them is at most $3 \gamma$. Thus, with probability at least $1-3 \gamma$, the chosen radius $R$ satisfies $R \leq R^{*}$, and the opening cost of the corresponding strategy is at most $(1+2 \epsilon) B$. Finally, for the returned strategy Theorem 5.2.11 holds as well, and the sample bound accounts for all iteration of Algorithm 10.

Additionally, we do not need fresh samples for each radius guess $R$; we can draw an appropriate number of samples $N$ upfront, and test all guesses in "parallel" with the same data.

## In light of Theorem 5.2.12 and the generic search step for the radius $R$, we assume for

 all our $\mathcal{P}$-poly problems that a target radius $R$ is given explicitly. We conclude with some final remarks. At first, $\mathbf{S 3}$ guarantees $N=\operatorname{poly}\left(n, m, \frac{1}{\epsilon}, \frac{1}{\alpha}, \log \frac{1}{\gamma}\right)$. Also, the probability $2 \alpha$ of not returning an $\eta$-approximate solution can be made inverse polynomially small, without affecting the polynomial nature of the sample complexity.
### 5.3 Solving 2S-Sup-Poly

In this section we tackle 2 S-SuP-BB, by first designing a 3-approximation algorithm for 2S-Sup-Poly and then proving that the latter is efficiently generalizable.

### 5.3.1 A 3-Approximation Algorithm for 2S-Sup-Poly

We are given a list of scenarios $Q$ together with their occurrence probabilities $p_{A}$ and second stage facility-cost vectors $c^{A}$, a target radius $R$, and let $G_{j}=G_{j, R}, i_{j}^{I}=i_{j, R}^{I}, i_{j}^{A}=i_{j, R}^{A}$ for every $j \in \mathcal{C}$ and $A \in Q$. Consider LP (5.4)-(5.6).

$$
\begin{align*}
& \sum_{i \in \mathcal{F}} y_{i}^{I} \cdot c_{i}^{I}+\sum_{A \in Q} p_{A} \sum_{i \in \mathcal{F}} y_{i}^{A} \cdot c_{i}^{A} \leq B  \tag{5.4}\\
& \sum_{i \in G_{j}}\left(y_{i}^{I}+y_{i}^{A}\right) \geq 1, \quad \forall j \in A \in Q  \tag{5.5}\\
& 0 \leq y_{i}^{I}, y_{i}^{A} \leq 1 \tag{5.6}
\end{align*}
$$

Constraint (5.4) captures the total expected cost, and constraint (5.5) the fact that for all $A \in Q$, every $j \in A$ must have an open facility within distance $R$ from it. In addition, note that if the LP is infeasible, then there cannot be a solution of radius at most $R$ for the given 2S-SuP-Poly instance. The rounding algorithm appears in Algorithm 11.

Theorem 5.3.1. For any scenario $A \in Q$ and every $j \in A$, we have $d\left(j, F_{I}^{\ell^{*}} \cup F_{A}^{\ell^{*}}\right) \leq 3 R$.

Proof. Focus on some $A \in Q$. Recall that $d\left(j, \pi^{I}(j)\right) \leq 2 R$ and $d\left(j, \pi^{A}(j)\right) \leq 2 R$ for any $j \in A$. For $j \in H_{A}$ the statement is clearly true, because either $G_{\pi^{I}(j)} \cap F_{I}^{\ell^{*}} \neq \emptyset$ or $G_{j} \cap F_{A}^{\ell^{*}} \neq \emptyset$. So consider some $j \in A \backslash H_{A}$. If $G_{\pi^{A}(j)} \cap F_{A}^{\ell^{*}} \neq \emptyset$, then any facility $i \in G_{\pi^{A}(j)} \cap F_{A}^{\ell^{*}}$ will be

```
Algorithm 11: Correlated LP-Rounding Algorithm for 2S-SuP-Poly
    Solve LP (5.4)-(5.6) to get a feasible solution \(y^{I}, y^{A}: A \in Q\);
    if no feasible LP solution exists then
        Return "INFEASIBLE";
    end
    \(\left(H_{I}, \pi^{I}\right) \leftarrow\) GreedyCluster \(\left(\mathcal{C}, R, g^{I}\right)\), where \(g^{I}(j)=y^{I}\left(G_{j}\right)\);
    for each scenario \(A \in Q\) do
        \(\left(H_{A}, \pi^{A}\right) \leftarrow \operatorname{GreedyCluster}\left(A, R, g^{A}\right)\), where \(g^{A}(j)=-y^{I}\left(G_{\pi^{I}(j)}\right) ;\)
    end
    Order the clients of \(H_{I}\) as \(j_{1}, j_{2}, \ldots, j_{h}\) such that \(y^{I}\left(G_{j_{1}}\right) \leq y^{I}\left(G_{j_{2}}\right) \leq \cdots \leq y^{I}\left(G_{j_{h}}\right)\);
    Consider a new "dummy" client \(j_{h+1}\) with \(y^{I}\left(G_{j_{h+1}}\right)>y^{I}\left(G_{j_{\ell}}\right)\) for all \(\ell \in[h]\);
    for all integers \(\ell=1,2, \ldots, h+1\) do
        \(F_{I}^{\ell} \leftarrow\left\{i_{j_{k}}^{I} \mid j_{k} \in H_{I}\right.\) and \(\left.y^{I}\left(G_{j_{k}}\right) \geq y^{I}\left(G_{j_{\ell}}\right)\right\} ;\)
        for each \(A \in Q\) do
            \(F_{A}^{\ell} \leftarrow\left\{i_{j}^{A} \mid j \in H_{A}\right.\) and \(\left.F_{I}^{\ell} \cap G_{\pi^{I}(j)}=\emptyset\right\} ;\)
        end
        \(S_{\ell} \leftarrow c^{I}\left(F_{I}^{\ell}\right)+\sum_{A \in Q} p_{A} \cdot c^{A}\left(F_{A}^{\ell}\right) ;\)
    end
    Return \(F_{I}^{\ell^{*}}, F_{A}^{\ell^{*}}: A \in Q\) such that \(\ell^{*}=\arg \min _{\ell} S_{\ell} ;\)
```

within distance $3 R$ from $j$. If on the other hand $G_{\pi^{A}(j)} \cap F_{A}^{\ell^{*}}=\emptyset$, then our algorithm guarantees $G_{\pi^{I}\left(\pi^{A}(j)\right)} \cap F_{I}^{\ell^{*}} \neq \emptyset$. Further, the stage-II greedy clustering yields

$$
g^{A}\left(\pi_{A}(j)\right) \geq g^{A}(j) \Longrightarrow y^{I}\left(G_{\pi^{I}(j)}\right) \geq y^{I}\left(G_{\pi^{I}\left(\pi^{A}(j)\right)}\right)
$$

Therefore, from the way we formed $F_{I}^{\ell^{*}}$ and the fact that $G_{\pi^{I}\left(\pi^{A}(j)\right)} \cap F_{I}^{\ell^{*}} \neq \emptyset$, we infer that $G_{\pi^{I}(j)} \cap F_{I}^{\ell^{*}} \neq \emptyset$. The latter ensures that $d\left(j, G_{\pi^{I}(j)} \cap F_{I}^{\ell^{*}}\right) \leq 3 R$.

Theorem 5.3.2. The opening cost $S_{\ell^{*}}$ of Algorithm 11 is at most $B$.

Proof. Consider the following process to generate a random solution: we draw a random variable $\beta$ uniformly from $[0,1]$, and then set $F_{I}^{\beta}=\left\{i_{j}^{I} \mid j \in H_{I}\right.$ and $\left.y^{I}\left(G_{j}\right) \geq \beta\right\}, F_{A}^{\beta}=\left\{i_{j}^{A} \mid j \in\right.$ $H_{A}$ and $\left.F_{I} \cap G_{\pi^{I}(j)}=\emptyset\right\}$ for all $A \in Q$. For each possible draw for $\beta$, the resulting sets $F_{I}^{\beta}, F_{A}^{\beta}$ correspond to sets $F_{I}^{\ell}, F_{A}^{\ell}$ for some integer $\ell \in[h+1]$. Hence, in order to show the existence of
$\ell$ with $S_{\ell} \leq B$, it suffices to show $\mathbb{E}_{\beta \sim[0,1]}\left[c^{I}\left(F_{I}^{\beta}\right)+\sum_{A \in Q} p_{A} \cdot c^{A}\left(F_{A}^{\beta}\right)\right] \leq B$.
We start by calculating the probability of opening a given facility $i_{j}^{I}$ with $j \in H_{I}$ in stageI. This will occur only if $\beta \leq y^{I}\left(G_{j}\right)$, and so $\operatorname{Pr}\left[i_{j}^{I}\right.$ is opened at stage-I $] \leq \min \left(y^{I}\left(G_{j}\right), 1\right)$. Therefore, due to $G_{j} \cap G_{j^{\prime}}=\emptyset$ for all distinct $j, j^{\prime} \in H_{I}$, we get:

$$
\begin{align*}
\mathbb{E}_{\beta \sim[0,1]}\left[c^{I}\left(F_{I}^{\beta}\right)\right] & \leq \sum_{j \in H_{I}} c_{i_{j}^{I}}^{I} \cdot y^{I}\left(G_{j}\right) \\
& \leq \sum_{i \in \mathcal{F}} y_{i}^{I} \cdot c_{i}^{I} \tag{5.7}
\end{align*}
$$

Moreover, for any $j \in H_{A}$ and any $A \in Q$ we have $\operatorname{Pr}\left[i_{j}^{A}\right.$ is opened at stage-II $\left.\mid A\right]=1-$ $\min \left(y^{I}\left(G_{\pi^{I}(j)}\right), 1\right) \leq 1-\min \left(y^{I}\left(G_{j}\right), 1\right) \leq y^{A}\left(G_{j}\right)$. The first inequality results from the greedy clustering of stage-I that gives $y^{I}\left(G_{\pi^{I}(j)}\right) \geq y^{I}\left(G_{j}\right)$, and the second follows from (5.5). Thus, due to $G_{j} \cap G_{j^{\prime}}=\emptyset$ for all distinct $j, j^{\prime} \in H_{A}$, we get:

$$
\begin{align*}
\mathbb{E}_{\beta \sim[0,1]}\left[c^{A}\left(F_{A}^{\beta}\right)\right] & \leq \sum_{j \in H_{A}} c_{i_{j}^{A}}^{A} \cdot y^{A}\left(G_{j}\right) \\
& \leq \sum_{i \in \mathcal{F}} y_{i}^{A} \cdot c_{i}^{A} \tag{5.8}
\end{align*}
$$

Combining (5.7), (5.8), (5.4) gives $\mathbb{E}_{\beta \sim[0,1]}\left[c^{I}\left(F_{I}^{\beta}\right)\right]+\sum_{A \in Q} p_{A} \cdot \mathbb{E}_{\beta \sim[0,1]}\left[c^{A}\left(F_{A}^{\beta}\right)\right] \leq B$.

### 5.3.2 Generalizing to the Black-Box Setting

To show that Algorithm 11 fits the framework of Section 5.2, we must show that it is efficiently generalizable as in Definition 5.2.4. For one thing, it is obvious that Algorithm 11 satisfies the properties of Definition 5.2.3, and therefore is a valid 3-approximation. Hence, we

```
Algorithm 12: Generalization Procedure for 2S-SuP-Poly
    Input: Returned sets \(F_{I}, F_{A}: A \in Q\) and inner execution details of Algorithm 11
    Let \(\bar{s}\) the strategy we will define, and for the stage-I actions set \(F_{I}^{\bar{s}} \leftarrow F_{I}\);
    Suppose scenario \(A \in \mathcal{D}\) arrived in the second stage;
    For every \(j \in A\) set \(g(j) \leftarrow-y^{I}\left(G_{\pi^{I}(j)}\right)\), where \(y^{I}\), \(\pi^{I}\) are the LP solution vector and
        stage-I mapping computed in Algorithm 11;
    \(\left(H_{A}, \pi^{A}\right) \leftarrow\) GreedyCluster \((A, R, g)\);
    \(F_{A}^{\bar{s}} \leftarrow\left\{i_{j}^{A} \mid j \in H_{A}\right.\) and \(\left.F_{I} \cap G_{\pi^{I}(j)}=\emptyset\right\} ;\)
```

only need a process to efficiently extend its output to any arriving scenario $A \in \mathcal{D}$, where $\mathcal{D}$ the black-box distribution. This is demonstrated in Algorithm 12, which mimics the stage-II actions of Algorithm 11. Here we crucially exploit the fact that the stage-II decisions of Algorithm 11 only depend on information from the LP about stage-I variables.

Since Algorithm 12 exactly imitates the stage-II actions of Algorithm 11, it is easy to see that property $\mathbf{S 2}$ is satisfied. Further, the arguments in Theorem 5.3.1 would still apply, and eventually guarantee $d\left(j, F_{I}^{\bar{s}} \cup F_{A}^{\bar{s}}\right) \leq 3 R$ for all $j \in A$ and any $A \in \mathcal{D}$, thus verifying property S1. We only need to prove $\mathbf{S 3}$. Let $\mathcal{S}_{K}$ the set of strategies achievable via Algorithm 12.

Lemma 5.3.3. Algorithm 11 satisfies property $\boldsymbol{S} 3$ with $\left|\mathcal{S}_{K}\right| \leq(n+1)!$.

Proof. The constructed final strategy is determined by 1) the sorted order of $y^{I}\left(G_{j}\right)$ for all $j \in \mathcal{C}$, and 2) a minimum threshold $\ell^{\prime}$ such that $G_{j_{\ell^{\prime}}} \cap F_{I} \neq \emptyset$ with $j_{\ell^{\prime}} \in H_{I}$. Given those, we know exactly what $H_{I}$ and $H_{A}$ for every $A \in \mathcal{D}$ will be, as well as $F_{I}$ and $F_{A}$ for every $A \in \mathcal{D}$. The set of all such options is independent of $Q$. Since there are $n!$ orderings for the $y^{I}\left(G_{j}\right)$ values, and the parameter $\ell^{\prime}$ takes at most $n+1$ values, $\left|\mathcal{S}_{K}\right| \leq(n+1)$ !.

### 5.4 Solving 2S-MatSup-Poly

We begin with a 5 -approximation algorithm for 2S-MATSUP-Poly, and then show that it is also efficiently generalizable so that we get results for 2 S-MATSUP-BB.

### 5.4.1 A 5-Approximation Algorithm for 2S-MATSuP-Poly

We are given $R$, and a list of scenarios $Q$ with their probabilities $p_{A}$ and cost vectors $c^{A}$. Also, let $r_{\mathcal{M}}$ be the rank function of the input matroid $\mathcal{M}=(\mathcal{F}, \mathcal{I})$. We again use the notation $G_{j}=G_{j, R}$, and $i_{j}^{A}=i_{j, R}^{A}$ for every $j \in \mathcal{C}$ and $A \in Q$. Consider LP (5.9)-(5.12).

$$
\begin{align*}
& \sum_{i \in \mathcal{F}} y_{i}^{I} \cdot c_{i}^{I}+\sum_{A \in Q} p_{A} \sum_{i \in \mathcal{F}} y_{i}^{A} \cdot c_{i}^{A} \leq B  \tag{5.9}\\
& \sum_{i \in G_{j}}\left(y_{i}^{I}+y_{i}^{A}\right) \geq 1, \quad \forall j \in A \in Q  \tag{5.10}\\
& \sum_{i \in U} y_{i}^{I} \leq r_{\mathcal{M}}(U), \quad \forall U \subseteq \mathcal{F}  \tag{5.11}\\
& 0 \leq y_{i}^{I}, y_{i}^{A} \leq 1 \tag{5.12}
\end{align*}
$$

Compared to LP (5.4)-(5.6), the only difference lies in constraint (5.11), which exactly represents the stage-I matroid requirement. Hence, it is a valid relaxation for the problem. Although the LP has an exponential number of constraints, it can be solved in polynomial time via the Ellipsoid algorithm, with a separation oracle based on minimizing a submodular function [128].

Assuming LP feasibility, our algorithm (presented in full detail in Algorithm 13), begins with two greedy clustering steps, one for each stage, that produce sets $H_{I}, H_{A}: A \in Q$ with corresponding mappings $\pi^{I}$ and $\pi^{A}$. We then set up and solve the auxiliary LP shown in (5.13)-

```
Algorithm 13: Rounding Algorithm for 2S-MATSuP-Poly
    Solve LP (5.9)-(5.12) to get a feasible solution \(y^{I}, y^{A}\) for all \(A \in Q\);
    if no feasible LP solution exists then
        Return "INFEASIBLE";
    end
    \(\left(H_{I}, \pi^{I}\right) \leftarrow \operatorname{GreedyCluster}\left(\mathcal{C}, R, g^{I}\right)\) where \(g^{I}(j)=y^{I}\left(G_{j}\right)\);
    Let \(g^{I I}: \mathcal{C} \mapsto[n]\) be some fixed and given bijective mapping;
    for each scenario \(A \in Q\) do
        \(\mid \quad\left(H_{A}, \pi^{A}\right) \leftarrow \operatorname{GreedyCluster}\left(A, R, g^{I I}\right)\);
    end
```

    Solve LP (5.13)-(5.16) and get an optimal integral solution \(z^{*}\), which will satisfy
        \(z_{i}^{*} \in\{0,1\}\) for all \(i \in \mathcal{F}\);
    \(F_{I} \leftarrow\left\{i \in \mathcal{F} \mid z_{i}^{*}=1\right\} ;\)
    \(F_{A} \leftarrow\left\{i_{j}^{A} \in \mathcal{F} \mid j \in H_{A}\right.\) and \(\left.G_{\pi^{I}(j)} \cap F_{I}=\emptyset\right\}\) for every \(A \in Q\).
    (5.16), and use this solution to determine sets $F_{I}$ and $F_{A}$.

$$
\begin{align*}
& \operatorname{minimize} \sum_{i \in \mathcal{F}} z_{i} \cdot c_{i}^{I}+\sum_{A \in Q} p_{A} \sum_{j \in H_{A}} c_{i_{j}^{A}}^{A}\left(1-z\left(G_{\pi^{I}(j)}\right)\right)  \tag{5.13}\\
& \text { subject to } z\left(G_{j}\right) \leq 1, \forall j \in H_{I}  \tag{5.14}\\
& z(U) \leq r_{\mathcal{M}}(U), \forall U \subseteq \mathcal{F}  \tag{5.15}\\
& 0 \leq z_{i} \leq 1 \tag{5.16}
\end{align*}
$$

Lemma 5.4.1. If $L P$ (5.9)-(5.12) is feasible, then the optimal solution $z^{*}$ of the auxiliary $L P$ (5.13)-(5.16) has objective function value at most $B$, and is integral $\left(z_{i}^{*} \in\{0,1\}\right.$ for all $\left.i \in \mathcal{F}\right)$.

Proof. Solution $z^{*}$ is integral since the LP (5.13)-(5.16) is the intersection of two matroid polytopes, namely, the polytope correspondind to $\mathcal{M}$, and a partition matroid polytope over all $G_{j}$ with $j \in H_{I}$. (Recall that sets $G_{j}$ for $j \in H_{I}$ are pairwise disjoint.)

Now let $y^{I}, y^{A}$ be a feasible solution of (5.9)-(5.12). For all $j \in H_{I}$ with $y^{I}\left(G_{j}\right) \leq 1$, set $z_{i}=y_{i}^{I}$ for all $i \in G_{j}$. For all $j \in H_{I}$ with $y^{I}\left(G_{j}\right)>1$, set $z_{i}=y_{i}^{I} / y^{I}\left(G_{j}\right)$ for all $i \in G_{j}$. For the
rest of the facilities set $z_{i}=0$. This solution obviously satisfies (5.14). Also, because $y^{I}$ satisfies (5.11) and $z_{i} \leq y_{i}^{I}$ for all $i$, we know that $z$ satisfies (5.15) too. Finally, regarding the objective:

$$
\begin{equation*}
\sum_{i \in \mathcal{F}} z_{i} \cdot c_{i}^{I} \leq \sum_{i \in \mathcal{F}} y_{i} \cdot c_{i}^{I} \tag{5.17}
\end{equation*}
$$

For the second-stage cost we then get:

$$
\begin{align*}
\sum_{A \in Q} p_{A} \sum_{j \in H_{A}} c_{i_{j}^{A}}^{A}\left(1-z\left(G_{\pi^{I}(j)}\right)\right) & \leq \sum_{A \in Q} p_{A} \sum_{\substack{j \in H_{A}: \\
y^{I}\left(G_{\pi^{I}(j)}\right) \leq 1}} c_{i_{j}^{A}}^{A}\left(1-y^{I}\left(G_{\pi^{I}(j)}\right)\right) \\
& \leq \sum_{A \in Q} p_{A} \sum_{\substack{j \in H_{A}: \\
y^{I}\left(G_{\pi^{I}(j)}\right) \leq 1}} c_{i_{j}^{A}}^{A}\left(1-y^{I}\left(G_{j}\right)\right) \\
& \leq \sum_{A \in Q} p_{A} \sum_{\substack{j \in H_{A}: \\
y^{I}\left(G_{\pi^{I}(j)}\right) \leq 1}} c_{i_{j}^{A}}^{A} y^{A}\left(G_{j}\right) \\
& \leq \sum_{A \in Q} p_{A} \sum_{i \in \mathcal{F}} y_{i}^{A} c_{i}^{A} \tag{5.18}
\end{align*}
$$

The second line follows from the stage-I greedy clustering, which ensures $y^{I}\left(G_{\pi^{I}(j)}\right) \geq y^{I}\left(G_{j}\right)$ for all $j \in \mathcal{C}$. The last line is due to (5.10), and the fact that for all $A \in Q$ and all distinct $j, j^{\prime} \in H_{A}$ we have $G_{j} \cap G_{j^{\prime}}=\emptyset$. Finally, combining (5.9), (5.17) and (5.18) proves that the opening cost of the returned solution is at most $B$.

Theorem 5.4.2. For the sets $F_{I}, F_{A}: A \in Q$ that constitute the solution returned by Algorithm 13 the following three properties hold: (i) $F_{I} \in \mathcal{I}$, (ii) $c^{I}\left(F_{I}\right)+\sum_{A \in Q} p_{A} c^{A}\left(F_{A}\right) \leq B$, and (iii) $d\left(j, F_{I} \cup F_{A}\right) \leq 5 R$ for all $j \in A \in Q$.

Proof. (i) is obvious since $z^{*}$ satisfies constraint (5.15). For (ii), the opening cost of the solution

```
Algorithm 14: Generalization Procedure for 2S-MATSUP-Poly
    Input: Returned sets \(F_{I}, F_{A}: A \in Q\) and inner execution details of Algorithm 13
    Let \(\bar{s}\) the strategy we will define, and for the stage-I actions set \(F_{I}^{\bar{s}} \leftarrow F_{I}\);
    Suppose scenario \(A \in \mathcal{D}\) arrived in the second stage;
    Let \(\pi^{I}\) the stage-I mapping and \(g^{I I}\) the bijective function, both used in Algorithm 13;
    Set \(\left(H_{A}, \pi^{A}\right) \leftarrow \operatorname{GreedyCluster}\left(A, R, g^{I I}\right)\);
    Open the set \(F_{A}^{\bar{s}}=\left\{i_{j}^{A} \mid j \in H_{A}\right.\) and \(\left.F_{I} \cap G_{\pi^{I}(j)}=\emptyset\right\}\);
```

coincides with the value of the objective (5.13) for $z^{*}$, and hence by Lemma 5.4.1 it is at most $B$.
For (iii), consider $A \in Q$, and recall that $d\left(j, \pi^{I}(j)\right) \leq 2 R$ and $d\left(j, \pi^{A}(j)\right) \leq 2 R$ for any $j \in A$. For $j \in H_{A}$ the bound (iii) holds, because either $G_{\pi^{I}(j)} \cap F_{I} \neq \emptyset$ or $G_{j} \cap F_{A} \neq \emptyset$. So suppose that $j \in A \backslash H_{A}$. If $G_{\pi^{A}(j)} \cap F_{A} \neq \emptyset$, then any facility $i \in G_{\pi^{A}(j)} \cap F_{A}$ will be within distance $3 R$ from $j$. If on the other hand $G_{\pi^{A}(j)} \cap F_{A}=\emptyset$, then there exists $i \in G_{\pi^{I}\left(\pi^{A}(j)\right)} \cap F_{I}$. Therefore, the triangle inequality in this case yields

$$
d(i, j) \leq d\left(i, \pi^{I}\left(\pi^{A}(j)\right)\right)+d\left(\pi^{I}\left(\pi^{A}(j)\right), \pi^{A}(j)\right)+d\left(\pi^{A}(j), j\right) \leq 5 R
$$

### 5.4.2 Generalizing to the Black-Box Setting

It is clear that Algorithm 13 satisfies Definition 5.2.3, and thus is a valid 5-approximation. Consider now Algorithm 14 to efficiently extend its output to any arriving scenario $A \in \mathcal{D}$. Since Algorithm 14 exactly imitates the stage-II actions of Algorithm 13, it is easy to see that property $\mathbf{S} \mathbf{2}$ is satisfied. Furthermore, the arguments in Theorem 5.4.2 would still go through, and eventually guarantee $d\left(j, F_{I}^{\bar{s}} \cup F_{A}^{\bar{s}}\right) \leq 5 R$ for all $j \in A$ and any $A \in \mathcal{D}$, thus verifying property S1. To conclude, we need to prove $\mathbf{S 3}$. Let $\mathcal{S}_{M}$ the set of strategies achievable via Algorithm 14.

Lemma 5.4.3. Algorithm 13 satisfies property $\boldsymbol{S 3}$ with $\left|\mathcal{S}_{M}\right|=2^{m} \cdot n!$.

Proof. Since $g^{I I}$ can be thought of as part of the input, $\bar{s}$ depends only on 1) the set $F_{I}$ returned
by Algorithm 13, and 2) the sorted order of $y^{I}\left(G_{j}\right)$ for all $j \in \mathcal{C}$, which ultimately dictates the mapping $\pi^{I}$. Given those, we can determine the stage-II openings for every possible scenario $A \in \mathcal{D}$. These options do not depend on scenarios $Q$. The total number of possible outcomes for $F_{I}$ is $2^{m}$, and the total number of orderings for the clients of $\mathcal{C}$ is $n!$. Hence, $\left|\mathcal{S}_{M}\right|=2^{m} \cdot n!$.

### 5.5 Solving 2S-MuSup-Poly

To tackle this, we construct an efficiently generalizable algorithm for 2S-SuP-Poly, via an intriguing reduction to a non-stochastic clustering problem with outliers. Specifically, if we view stage-I as consisting of a deterministic robust problem, stage-II can be interpreted as covering all outliers left over by stage-I. Formally, we use the following robust supplier problem:

Definition 5.5.1 (Robust Weighted Multi-Knapsack-Supplier). We are given a set of clients $\mathcal{C}$ and a set of facilities $\mathcal{F}$, in a metric space with distance function $d$. The input also includes parameters $V, R \in \mathbb{R}_{\geq 0}$, and for every client $j \in \mathcal{C}$ an associated weight $v_{j} \in \mathbb{R}_{\geq 0}$. In addition, we have the same types of multi-knapsack constraints as in 2S-MuSup: there are $L$ in total budgets $W_{\ell}$, and every facility $i \in \mathcal{F}$ has costs $f_{i}^{\ell}$ for $\ell \in[L]$. The goal is to choose a set of facilities $S \subseteq \mathcal{F}$, such that $\sum_{j \in \mathcal{C}: d(j, S)>R} v_{j} \leq V$ and $f^{\ell}(S) \leq W_{\ell}$ for every $\ell \in[L]$. Clients $j$ with $d(j, S)>R$ are called outliers. An instance of this problem is called discrete, if the values $f_{i}^{\ell}$ are all integers.

We first show that any $\rho$-approximation algorithm for Robust Weighted Multi-KnapsackSupplier can be used in order to get an efficiently generalizable ( $\rho+2$ )-approximation algorithm for 2S-MuSup-Poly. In addition, we argue that already existing work [129, 130] gives a 3approximation for discrete instances of Robust Weighted Multi-Knapsack-Supplier, thus leading to an efficiently generalizable 5-approximation for discrete instances of 2S-MUSUP-Poly.

### 5.5.1 Reducing 2S-MuSup-Poly to Robust Weighted Multi-Knapsack-Sup

We first suppose that the costs $c_{i}^{I}$ are polynomially bounded integers; this restriction will be removed when we generalize to the black-box setting. Once more, let $Q$ be a set of provided scenarios, $R$ a target radius, and $G_{j}=G_{j, R}, i_{j}^{A}=i_{j, R}^{A}$ for all $j \in \mathcal{C}$ and $A \in Q$. Furthermore, suppose that we have a $\rho$-approximation $R W$ for Robust Weighted Multi-Knapsack-Supplier. For a feasible instance $\mathfrak{I}^{\prime}$ of the latter problem, $R W$ returns a solution $S$ satisfying all knapsack constraints and also $\sum_{j \in \mathcal{C}: d(j, S)>\rho R} v_{j} \leq V$. Otherwise, it either returns "INFEASIBLE", or again a solution with the previous properties.

If the provided instance $\mathfrak{I}$ of 2 S-MUSUP-Poly is feasible, the first step in tackling the problem is figuring out the portion of the budget, say $B_{I}$, that is used in the first stage of a feasible solution. Since the costs $c_{i}^{I}$ are polynomially bounded integers, we can guess $B_{I}$ in polynomial time through solving the problem for all different alternatives for it. So from this point on, assume w.l.o.g. that we have the correct $B_{I}$, and also let $B_{I I}=B-B_{I}$.

Algorithm 15 shows how to use $R W$ to approximate 2 S-MuSup-Poly. It begins with greedy clustering steps for each $A$, and given $H_{A}, \pi^{A}$ it constructs an instance $\mathfrak{I}^{\prime}$ of Robust Weighted Multi-Knapsack-Supplier as follows. $\mathcal{C}, \mathcal{F}, d$, and $R$ are the same for both problems.

For all $j \in \mathcal{C}$ we set

$$
v_{j}=\sum_{A \in Q: j \in H_{A}} p_{A} \cdot c_{i_{j}^{A}}^{A}
$$

and also $V=B_{I I}$. Finally, the instance $\mathfrak{I}^{\prime}$ has $L^{\prime}=L+1$ knapsack constraints, where the first $L$ are the stage-I constraints of 2S-MUSUP-Poly $\left(f^{\ell}(S) \leq W_{\ell}\right)$, and the last is $c^{I}(S) \leq B_{I}$.

Lemma 5.5.2. If the original 2 S-MuSup-Poly instance $\mathfrak{I}$ is feasible, then the Robust Weighted

```
Algorithm 15: Approximation Algorithm for 2S-MUSUP-Poly
    Let \(g^{I I}: \mathcal{C} \mapsto[n]\) be some fixed and given bijective mapping;
    for each scenario \(A \in Q\) do
    \(\mid \quad\left(H_{A}, \pi^{A}\right) \leftarrow \operatorname{GreedyCluster}\left(A, R, g^{I I}\right)\);
    end
    Construct instance \(\mathfrak{I}^{\prime}\) of Robust Weighted Multi-Knapsack-Supplier as discussed;
    if \(R W\left(\mathfrak{I}^{\prime}\right)=\) "INFEASIBLE" then
        Return "INFEASIBLE";
    end
    \(F_{I} \leftarrow R W\left(\mathfrak{I}^{\prime}\right) ; \quad / /\) Stage-I facilities
    for each scenario \(A \in Q\) do
        \(F_{A} \leftarrow\left\{i_{j}^{A} \mid j \in H_{A}\right.\) with \(\left.d\left(j, F_{I}\right)>\rho R\right\} ; \quad / /\) Stage-II facilities
    end
```

Multi-Knapsack-Supplier instance $\mathfrak{I}^{\prime}$ is also feasible.

Proof. Consider some feasible solution $F_{I}^{\star}, F_{A}^{*}$ for 2S-MUSUP-Poly. We claim that $F_{I}^{\star}$ is a valid solution for $\mathfrak{I}^{\prime}$. It clearly satisfies the $L$ knapsack constraints of the form $f^{\ell}\left(F_{I}^{*}\right) \leq W_{\ell}$, and if our guess $B_{I}$ is the right one, it also satisfies $c^{I}\left(F_{I}^{\star}\right) \leq B_{I}$. Now, for any $A \in Q$, any client $j \in H_{A}$ with $d\left(j, F_{I}^{\star}\right)>R$ must be covered by some facility $x_{j}^{A} \in G_{j} \cap F_{A}^{\star}$. Since $B_{I I}$ is the stage-II portion of the budget used by $F_{I}^{\star}, F_{A}^{*}$ and $G_{j^{\prime}} \cap G_{j^{\prime \prime}}=\emptyset$ for all distinct $j^{\prime}, j^{\prime \prime} \in H_{A}$, we have:

$$
\begin{aligned}
B_{I I} & \geq \sum_{A} p_{A} \sum_{i \in F_{A}^{\star}} c_{i}^{A} \\
& \geq \sum_{A} p_{A} \sum_{\substack{j \in H_{A}: \\
d\left(j, F_{I}^{*}\right)>R}} c_{x_{j}^{A}}^{A} \\
& \geq \sum_{A} p_{A} \sum_{\substack{j \in H_{A}: \\
d\left(j, F_{I}^{*}\right)>R}} c_{i_{j}^{A}}^{A} \\
& =\sum_{\substack{j \in \mathcal{C} \\
d\left(j, F_{I}^{*}\right)>R}} v_{j}
\end{aligned}
$$

This implies that $S=F_{I}^{\star}$ satisfies the constraint $\sum_{j: d(j, S)>R} v_{j} \leq B_{I I}$ of instance $\mathfrak{I}^{\prime}$.

Theorem 5.5.3. Algorithm 15 is a valid ( $\rho+2$ )-approximation for 2S-MUSUP-Poly.

Proof. First of all, Lemma 5.5.2 guarantees that if the given instance of 2S-MuSup-Poly is feasible, we will get a solution $F_{I}, F_{A}$. By specification of $R W, c^{I}\left(F_{I}\right) \leq B_{I}$ and $f^{\ell}\left(F_{I}\right) \leq W_{\ell}$ for every $\ell$. The stage-II cost $C_{I I}$ of this solution is given by:

$$
\begin{aligned}
C_{I I} & =\sum_{A} p_{A} \sum_{\substack{j \in H_{A}: \\
d\left(j, F_{I}\right)>\rho R}} c_{i_{j}^{A}}^{A} \\
& =\sum_{\substack{j \in \mathcal{C} \\
d\left(j, F_{I}\right)>\rho R}} v_{j} \\
& \leq B_{I I},
\end{aligned}
$$

where the last inequality follows because $F_{I}$ is the output of $R W\left(\mathfrak{I}^{\prime}\right)$.
Consider now a $j \in A$ for some $A \in Q$. The distance of $j$ to its closest facility will be at $\operatorname{most} d\left(\pi^{A}(j), F_{I} \cup F_{A}\right)+d\left(j, \pi^{A}(j)\right)$. Since $\pi^{A}(j) \in H_{A}$, there will either be a stage-I open facility within distance $\rho R$ from it, or we perform a stage-II opening in $G_{\pi(j)}$, which results in a covering distance of at most $R$. Also, by the greedy clustering step, we have $d\left(j, \pi^{A}(j)\right) \leq 2 R$. So in the end we get $d\left(j, F_{I} \cup F_{A}\right) \leq(\rho+2) R$.

By combining Algorithm 15 with existing 3-approximations for Robust Weighted Multi-Knapsack-Supplier, we get the following result:

Theorem 5.5.4. There is a 5-approximation algorithm for discrete instances of 2S-MUSUP-Poly, where additionally all $c_{i}^{I}$ are polynomially bounded integers. The runtime of it is $\operatorname{poly}(n, m, \Lambda)$.

Proof. The results of Chakrabarty and Negahbani [129] give a 3-approximation for discrete instances of Robust Weighted Multi-Knapsack-Supplier, when $v_{j}=1$ for all $j$. The work of

```
Algorithm 16: Generalization Procedure for 2S-MUSUP-Poly
    Input: Returned sets \(F_{I}, F_{A}: A \in Q\) and inner execution details of Algorithm 15
    Let \(\bar{s}\) the strategy we will define, and for the stage-I actions set \(F_{I}^{\bar{s}} \leftarrow F_{I}\);
    Suppose scenaio \(A\) arrived in the second stage;
    \(\left(H_{A}, \pi^{A}\right) \leftarrow\) GreedyCluster \(\left(A, R, g^{I I}\right)\), where \(g^{I I}\) the bijective function used in
    Algorithm 15;
    Open the set \(F_{A}^{\bar{s}} \leftarrow\left\{i_{j}^{A} \mid j \in H_{A}\right.\) and \(\left.d\left(j, F_{I}\right)>\rho R\right\} ;\)
```

Pietracaprina et al. [130] extends this to allow arbitrary $v_{j}$ values. Note that by our assumption that the values $c^{I}$ are polynomially bounded integers, the instance $\mathfrak{I}^{\prime}$ is discrete, and hence the algorithm of Pietracaprina et al. [130] can be utilized in Algorithm 15 and give a 5-approximation for 2S-MuSup-Poly. Finally, given the results in [129, 130], the runtime of the whole process will be poly $(n, m, \Lambda)$.

### 5.5.2 Generalizing to the Black-Box Setting

Since the algorithm of Section 5.5.1 is a valid $(\rho+2)$-approximation one, we only need to demonstrate an extension procedure for it. Hence, consider the process shown in Algorithm 16, which efficiently extends the output of the previous polynomial-scenarios algorithm to any arriving scenario $A \in \mathcal{D}$.

Because Algorithm 16 exactly mimics the stage-II actions of Algorithm 15, it is easy to see that property $\mathbf{S} \mathbf{2}$ is satisfied. Moreover, the arguments of Theorem 5.5.3 would still be applicable and ensure $d\left(j, F_{I} \cup F_{A}\right) \leq(\rho+2) R$ for every $j \in A$ and $A \in \mathcal{D}$, thus guaranteeing property $\mathbf{S 1}$. To conclude, we again only need to prove property $\mathbf{S 3}$. Let $\mathcal{S}_{M K}$ the set of strategies achievable via Algorithm 16. We show the following.

Lemma 5.5.5. Algorithm 15 satisfies property $\boldsymbol{S} \mathbf{3}$ with $\left|\mathcal{S}_{M K}\right|=2^{m}$.

Proof. The returned final strategy depends solely on the set $F_{I}$. Given that, we can exactly
determine all possible stage-II openings, since every $H_{A}$ for $A \in \mathcal{D}$ can be computed using the fixed function $g^{I I}$. There are $2^{m}$ choices for $F_{I}$, and therefore $\left|\mathcal{S}_{M K}\right|=2^{m}$. Finally, it is easy to see that the set $\mathcal{S}_{M K}$ is independent of $Q$.

Our algorithm for 2S-MuSUP-Poly requires the values $c_{i}^{I}$ to be polynomially bounded integers. As we show next, this assumption can be removed by a standard rescaling trick:

Theorem 5.5.6. Suppose that the $c_{i}^{I}$ are arbitrary numbers. Through a standard cost-quantization technique that works for any $\epsilon \in(0,1)$, Algorithm 15 can be modified to give a solution $F_{I}, F_{A}$ : $A \in Q$ for 2S-MuSup-Poly, where $d\left(j, F_{I} \cup F_{A}\right) \leq(\rho+2) R$ for all $A \in Q, j \in A$, and also $c^{I}\left(F_{I}\right)+\sum_{A \in Q} p_{A} c^{A}\left(F_{A}\right) \leq(1+\epsilon) B$.

Proof. For convenience, let us assume that $B=1$, and suppose that all facilities have $c_{i}^{I} \leq B=1$ (as otherwise they can never be opened). Given some $\epsilon>0$, let us define $q=\epsilon / m$, and form new costs by $\tilde{c}_{i}^{I}=\left\lceil c_{i}^{I} / q\right\rceil, \tilde{c}_{i}^{A}=c_{i}^{A} / q, B^{\prime}=B(1+\epsilon) / q$. The costs $\tilde{c}_{i}^{I}$ are at most $\lceil 1 / q\rceil$, and hence are polynomially-bounded integers. Therefore, the reduction of Section 5.5.1 can be applied.

Suppose now that $F_{I}, F_{A}$ is a solution to the original instance of 2S-MUSUP-Poly, with opening cost at most $B$. For the modified cost of this solution we then have:

$$
\begin{aligned}
\tilde{c}^{I}\left(F_{I}\right)+\sum_{A} p_{A} \tilde{c}^{A}\left(F_{A}\right) & \leq\left(c^{I}\left(F_{I}\right)+\sum_{A} p_{A} c^{A}\left(F_{A}\right)\right) / q+\sum_{i \in \mathcal{F}} 1 \\
& \leq B / q+m \leq B^{\prime}
\end{aligned}
$$

Thus, $F_{I}, F_{A}$ is also a solution to the modified instance, implying that the latter is feasible. Hence, consider any solution $\tilde{F}_{I}, \tilde{F}_{A}$ to the modified instance, that we would get after running Algorithm

15 with the new costs; its opening cost in the original instance is

$$
\begin{aligned}
c^{I}\left(\tilde{F}_{I}\right)+\sum_{A} p_{A} c^{A}\left(\tilde{F}_{A}\right) & \leq q \tilde{c}^{I}\left(\tilde{F}_{I}\right)+q \sum_{A} p_{A} \tilde{c}^{A}\left(\tilde{F}_{A}\right) \\
& \leq q B^{\prime} \\
& =B(1+\epsilon)
\end{aligned}
$$

Therefore, since $\tilde{F}_{I}, \tilde{F}_{A}$ is a $(\rho+2)$-approximate solution, we get the desired result.

Note that applying our generalization framework on this solution would make the overall cost over $\mathcal{D}$ be at most $(1+\mathcal{O}(\epsilon))(1+\epsilon) B=(1+\mathcal{O}(\epsilon)) B$, which implies that in the black-box setting we do not need the initial assumption for the $\operatorname{costs} c_{i}^{I}$.

### 5.5.3 Connections to 2S-MATSUP-Poly

Suppose we define our non-stochastic robust problem as having one knapsack and one matroid constraint, instead of $L$ knapsack constraints. Then the reduction of Section 5.5.1 would yield a $(\rho+2)$-approximation for 2S-MATSUP-Poly in the exact same manner, where $\rho$ the ratio of the algorithm used to solve the corresponding deterministic outliers problem.

A result of Chakrabarty and Negahbani [129, Theorem 16] gives a 3-approximation for this outliers problem, which in turn would give a 5 -approximation for 2 S-MATSUP-Poly. However, the algorithm obtained in this way would be randomized (its solution may not be a valid one), would only work for polynomially bounded values $v_{j}$, and would also be significantly more complex than the algorithm of Section 5.4.

### 5.6 Applying the Standard SAA Method in Supplier Problems

Consider the standard two-stage stochastic setting. In the first stage, we are allowed to take some proactive actions and commit to an anticipatory part of the solution $x$, which will incur some cost $c(x)$. In the second stage, a scenario $A$ is sampled from the distribution $\mathcal{D}$, and we can take some stage-II recourse actions $y_{A}$ with cost $f_{A}\left(x, y_{A}\right)$. If $X$ is the set of stage-I actions and $Y$ the set of recourse actions, the goal is to find a solution $x^{\star} \in X$ to minimize $f(x)=c(x)+\mathbb{E}_{A \sim \mathcal{D}}\left[q_{A}(x)\right]$, where $q_{A}(x)=\min _{y \in Y}\left\{f_{A}(x, y) \mid(x, y)\right.$ is a valid solution for $\left.A\right\}$.

### 5.6.1 The Standard SAA Method:

Consider minimizing $f(x)$ in the black-box model. If $S$ is a set of scenarios sampled from the black-box oracle, let $\hat{f}(x)=c(x)+\left(\sum_{A \in S} q_{A}(x)\right) /|S|$ be the empirical estimate of $f(x)$. Also, let $x^{*}$ and $\bar{x}$ be the minimizers of $f(x)$ and $\hat{f}(x)$ respectively.

The work Swamy and Shmoys [60] shows that if $f(x)$ is modeled as a convex program, then for any $\epsilon, \gamma \in(0,1)$ and with $|S|=\operatorname{poly}(n, m, \lambda, \epsilon, 1 / \gamma)$, we have $f(\bar{x}) \leq(1+\epsilon) f\left(x^{*}\right)$ with probability at least $1-\gamma$ ( $\lambda$ is the maximum multiplicative factor by which an element's cost is increased in stage-II). An alternate proof of this appeared in [63], which also covered the case of $f(x)$ being an integer program. Moreover, Charikar et al. [63] prove that if $\bar{x}$ is an $\alpha-$ approximate minimizer of $\hat{f}(x)$, then a slight modification to their sampling approach still gives $f(\bar{x}) \leq(\alpha+\epsilon) f\left(x^{*}\right)$ with probability at least $1-\gamma$.

The result of Charikar et al. [63] further implies that the black-box model can be effectively reduced to the polynomial-scenarios one, via the following process. Assuming that $f(x)$ corresponds to the integer program modeling our problem, first find an $\alpha$-approximate minimizer $\bar{x}$ of
$\hat{f}(x)$, and treat $\bar{x}$ as the stage-I actions. Then, given any arriving $A$, re-solve the problem using any known $\rho$-approximation algorithm for the non-stochastic counterpart, with $\bar{x}$ as a fixed part of the solution. This process eventually leads to an overall approximation ratio of $\alpha \rho+\epsilon$.

### 5.6.2 Roadblocks for the Standard SAA Analysis in Supplier Problems

A natural way to fit the problems we studied here within existing SAA frameworks, is to first assume knowledge of the optimal radius $R^{*}$ and then use the opening cost as the objective function $f_{R^{*}}(x)$, by turning the radius requirement into a simple covering constraint. In other words, set $f_{R^{*}}(x)=c^{I}(x)+\mathbb{E}_{A \sim \mathcal{D}}\left[q_{A, R^{*}}(x)\right]$ with $q_{A, R^{*}}(x)=\min _{y}\left\{c^{A}(y) \mid(x, y)\right.$ covers all $j \in$ $A$ within distance $\left.R^{*}\right\}$. Note that $f_{R^{*}}(x)$ may represent both the convex and the integer program corresponding to the underlying problem.

To avoid any overhead in the approximation ratio (from re-solving the problem in stageII as suggested by the approach of Charikar et al. [63]), one should apply SAA to the function $f_{R^{*}}(x)$ corresponding to the convex program describing the problem (the roadblock described here trivially extends to the case of $f_{R^{*}}(x)$ being an integer function as well). If there exists a rounding that turns the empirical minimizer $\bar{x}_{R^{*}}$ into a solution that covers each client within distance $\alpha R^{*}$, while also having an opening cost of at most $f_{R^{*}}\left(\bar{x}_{R^{*}}\right)$, we get the desired result because $f_{R^{*}}\left(\bar{x}_{R^{*}}\right) \leq(1+\epsilon) f_{R^{*}}\left(x_{R^{*}}^{*}\right)$ and $f_{R^{*}}\left(x_{R^{*}}^{*}\right) \leq B$. With slight modifications, all our polynomial-scenarios algorithms can be interpreted as such rounding procedures.

Nonetheless, we still have to identify a good guess for $R^{*}$, and this constitutes an unavoidable roadblock in applying standard SAA in supplier problems. Since $R^{*}$ is one of $n m$ alternative options, one can test each of those individually. Hence, assume we work with some
guess $R$, and define the corresponding cost functions $f_{R}, \hat{f}_{R}$ with minimizers $x_{R}^{*}, \bar{x}_{R}$ respectively. Observe that $R$ is a good guess iff $f_{R}\left(x_{R}^{*}\right) \leq(1+\mathcal{O}(\epsilon)) B$, since in this way vanilla SAA combined with our rounding procedures yields an opening cost of $f_{R}\left(\bar{x}_{R}\right) \leq(1+\epsilon) f_{R}\left(x_{R}^{*}\right)$, and minimizing over the radius is just a matter of finding the minimum good guess. However, because $f_{R}(x)$ is not efficiently computable, the only way to test if $R$ is a good guess, is through $\hat{f}_{R}(x)$. Unfortunately, empirically estimating $f_{R}(x)$ within an $(1+\epsilon)$ factor may require a superpolynomial number of samples [57]. The reason for this is the existence of scenarios with high stage-II cost appearing with small probability, which drastically increase the variance of $\hat{f}_{R}(x)$. On a high level, the obstacle in supplier problems stems from the need to not only find a minimizer $\bar{x}_{R}$, but also compute its corresponding value $f_{R}\left(\bar{x}_{R}\right)$. This makes it impossible to know which guesses $R$ are good, and consequently there is no way to optimize over the radius.

Finally, note that if the stage-II cost of every scenario is polynomially bounded, the variance of $\hat{f}_{R}(x)$ is also polynomial, and standard SAA arguments would go through without difficulties (everything can easily handled with standard Chernof/Hoeffding bound). However, this assumption is much stronger than what is typically used for the two-stage stochastic model.

## Chapter 6: Conclusions and Future Work

The goal of the research presented in this dissertation was to generalize classical computational problems, e.g., clustering and graph-cut problems, along two directions, and then provide algorithms with provable guarantees for the novel variants.

The first direction involved incorporating considerations of fairness in our problems of interest. Towards that goal, in the context of $k$-clustering, we introduced two new notions of fairness, one yielding PairFairclu [37,38] and the other one yielding EQCenter [46]. Furthermore, we proposed the first ever fair graph-cut problems [53], by incorporating two already established concepts of fairness into SB-MinCC [13]; when considering demographic fairness this resulted in DEMFAIRCUT, and when considering probabilistic individual fairness that gave rise to IndFaircut. For all the previously mentioned problems we provided approximation algorithms with theoretical guarantees.

As for the second research direction that we followed, that was incorporating elements of stochasticity in classical problems. A first step towards this was studying PairFairClu, where we experimented with how randomness can guarantee fairness. In addition, motivated by ways to prevent or mitigate the spread of a contagious disease, we studied MinInfEDGE and 2S-Sup. The former can be interpreted as a stochastic graph-cut problem, while the latter is a two-stage stochastic generalization of a clustering/facility-location setting.

### 6.1 Future Work

Regarding PairFairclu, recall that our results from [37] only work for a specific notion of similarity. On the other hand, our result from [38] can handle arbitrary similarity functions, however its is only a pseudo-approximation result. This is because the BSP constraints are violated by a multiplicative factor of 2 . The important open question here is can we devise a true approximation algorithm for PAIRFAIRCLU?

As for EQCENTER, an interesting direction is considering $k$-median (or $k$-means) under our newly introduced constraints (1.2), (1.1). Since the feasibility result for $\alpha \geq 2$ holds for all $k$-clustering objectives, we know that the previous question is actually well-defined.

Moving on to our fair cut problems and specifically to DEMFAIRCUT, the question of whether or not there exists a true approximation for it when $\gamma$ is not a constant, remains open; recall that in our algorithm there is an $(1-\epsilon)$ multiplicative violation in the covering constraints. Another potential direction is breaching the gap between the approximation factor of $O(\log n$. $\log \gamma$ ) that we give for general $\gamma$, and the lower bound of $\log \gamma$ shown in Corollary 3.2.12.

Despite the fundamental nature of MinInFEDGE, its computational complexity remained open for the $p<1$ setting. A number of heuristics have been proposed, and rigorous algorithms are only known for very special random graphs. In our work we presented the first rigorous approximation results for this problem for certain classes of instances; however, even these turn out to be quite challenging, and require adapting the cut sparsification and sample-average approximation techniques in a non-trivial manner. Our work raises several interesting questions. First, it would be interesting to extend the result based on Karger's cut sparsification technique to the non-uniform probability setting. Second, it would be interesting to extend our work to
other realistic random models of social-contact networks, and to also identify what reasonable assumptions on deterministic network models would guarantee efficient solutions.

In the two-stage stochastic problems studied here, the goal was to minimize the maximum assignment distance for all points, subject to a stochastic budget constraint. Our intention is to study the same setting again, but this time under a different objective function, and more specifically the $k$-median one. That is, we would like to minimize

$$
\sum_{A} p_{A} \sum_{j \in A} d\left(j, F_{I} \cup F_{A}\right)
$$

where as usual $d(j, S)=\min _{i \in S} d(i, j)$.
Besides open questions pertaining to the problems already studied in this work, there are two more intriguing avenues of future research that we would like to mention.

Making the study of fairness an interdisciplinary area: Incorporating considerations of fairness in computational problems is not an issue that should be taken lightly by researchers, and this is obviously because of the profound impact automated decision making has in our everyday lives. There exist studies showing that computational approaches which were developed so that they satisfy fairness constraints, ended up causing more harm to the corresponding marginalized groups [131, 132]. This is clearly not because the researchers who developed these approaches had ill intentions. We rather believe that such instances occur because the nature of fairness considerations is too delicate for computer scientists alone to be able to address them. Therefore, computer scientists developing "fair" algorithms should work hand in hand with people from social sciences, civil rights groups, and of course the stakeholders; it is those that get affected by our solutions that truly appreciate their consequences.

Combining fairness and explainability: An emerging discipline in machine learning is explainability; the search of solutions that do not only efficiently optimize some goal, but are also easily explainable to non-expert stakeholders. A canonical example demonstrating the significance of this area, can be found when considering banks using ML systems to come up with loan approvals. In such cases, the bank should be able to explain to an applicant why they were not approved for a loan. Hence, an ML approach that produces solutions that are too obscure and complicated cannot be helpful towards such a goal. A very intriguing research direction revolving around such issues is combining fairness with explainability. Specifically, using concepts of fairness that are easily accessible/explainable to non-experts. For instance, is a stochasticity-based fairness concept really explainable and convincing to someone with no knowledge of statistics?

## Appendix A: Additional Experimental Results for EQCENTER

## Experimental results for Bank:



Figure A.1: Bank: Maximum assignment distance for all algorithms


Figure A.2: Bank: Satisfaction of fairness constraints


Figure A.3: Bank: Amount of constraint violation

## Experimental results for Creditcard:



Figure A.4: Credit: Maximum assignment distance for all algorithms


Figure A.5: Credit: Satisfaction of fairness constraints


Figure A.6: Credit: Amount of constraint violation

## Experimental results for Census1990:



Figure A.7: Census: Maximum assignment distance for all algorithms


Figure A.8: Census: Satisfaction of fairness constraints


Figure A.9: Census: Amount of constraint violation

## Experimental results for Diabetes:



Figure A.10: Diabetes: Maximum assignment distance for all algorithms


Figure A.11: Diabetes: Satisfaction of fairness constraints


Figure A.12: Diabetes: Amount of constraint violation

## Bibliography

[1] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. ACM Comput. Surv., 54(6), July 2021.
[2] Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness and Machine Learning. fairmlbook.org, 2019. http://www. fairmlbook.org.
[3] Emily Badger. How airbnb plans to fix its racial-bias problem. The Washington Post, 2016. September 8, 2016.
[4] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings. Proc. Priv. Enhancing Technol., 2015(1):92-112, 2015.
[5] Katie Benner, Glenn Thrush, and Mike Isaac. Facebook engages in housing discrimination with its ad practices. The New York Times, 2019. March 28, 2019.
[6] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ITCS '12, 2012.
[7] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. In Proceedings on Privacy Enhancing Technologies, pages 92-112, 2015.
[8] Dorit S. Hochbaum and David B. Shmoys. A best possible heuristic for the k-center problem. Math. Oper. Res., 10(2), May 1985.
[9] Christina Ilvento. Metric learning for individual fairness, 2019. URL https://arxiv. org/abs/1906.00250.
[10] Christopher Jung, Michael Kearns, Seth Neel, Aaron Roth, Logan Stapleton, and Zhiwei Steven Wu. An algorithmic framework for fairness elicitation, 2019. URL https: //arxiv.org/abs/1905.10660.
[11] Preethi Lahoti, Krishna P. Gummadi, and Gerhard Weikum. Operationalizing individual fairness with pairwise fair representations. Proc. VLDB Endow., 13(4):506-518, dec 2019. ISSN 2150-8097. doi: 10.14778/3372716.3372723. URL https://doi.org/10. 14778/3372716.3372723.
[12] Ara Hayrapetyan, David Kempe, Martin Pál, and Zoya Svitkina. Unbalanced graph cuts. In Proceedings of the 13th Annual European Conference on Algorithms, ESA'05, page 191-202, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3540291180.
[13] Zoya Svitkina and Éva Tardos. Min-max multiway cut. In Klaus Jansen, Sanjeev Khanna, José D. P. Rolim, and Dana Ron, editors, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, pages 207-218, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
[14] Stephen Eubank, Hasan Guclu, V S Anil Kumar, Madhav V Marathe, Aravind Srinivasan, Zoltán Toroczkai, and Nan Wang. Modelling disease outbreaks in realistic urban social networks. Nature, 429(6988):180—184, May 2004. ISSN 0028-0836. doi: 10.1038/ nature 02541 .
[15] David G. Harris, Thomas Pensyl, Aravind Srinivasan, and Khoa Trinh. A Lottery Model for Center-Type Problems with Outliers. In Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017), 2017.
[16] Georg Anegg, Haris Angelidakis, Adam Kurpisz, and Rico Zenklusen. A technique for obtaining true approximations for k-center with covering constraints. In Daniel Bienstock and Giacomo Zambelli, editors, Integer Programming and Combinatorial Optimization, 2020.
[17] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. Reviews of modern physics, 87 (3):925, 2015.
[18] M. Marathe and A. Vullikanti. Computational epidemiology. Communications of the ACM, 56(7):88-96, 2013.
[19] Stephen G. Eubank, V. S. Anil Kumar, Madhav V. Marathe, Aravind Srinivasan, and Nan Wang. Structural and algorithmic aspects of massive social networks. In J. Ian Munro, editor, Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004, pages 718-727. SIAM, 2004.
[20] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. Science, 286(5439):509-512, 1999.
[21] B. Bollobás and O. Riordan. Robustness and vulnerability of scale-free random graphs. Internet Mathematics, 2004.
[22] Fan R. K. Chung and Linyuan Lu. The volume of the giant component of a random graph with given expected degrees. SIAM J. Discret. Math., 20(2):395-411, 2006.
[23] E. M. L. Beale. On minimizing a convex function subject to linear inequalities. Journal of the Royal Statistical Society. Series B (Methodological), pages 173-184, 1955.
[24] George B. Dantzig. Linear programming under uncertainty. Management Science, pages 197-206, 1955.
[25] Chaitanya Swamy and David B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. SIGACT News, pages 33-46, 2006.
[26] Chaitanya Swamy. Risk-averse stochastic optimization: Probabilistically-constrained models and algorithms for black-box distributions: (extended abstract). Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1627-1646, 2011.
[27] Anupam Gupta, Martin Pál, R. Ravi, and Amitabh Sinha. Boosted sampling: Approximation algorithms for stochastic optimization. In Proc. 36th Annual ACM Symposium on Theory of Computing, STOC '04, page 417-426, 2004.
[28] David Shmoys and Chaitanya Swamy. An approximation scheme for stochastic linear programming and its application to stochastic integer programs. J. ACM, 53:978-1012, 11 2006.
[29] Andre Linhares and Chaitanya Swamy. Approximation algorithms for distributionallyrobust stochastic optimization with black-box distributions. In Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, pages 768-779, 2019.
[30] Chaitanya Swamy and David Shmoys. Sampling-based approximation algorithms for multistage stochastic optimization. pages $357-366,2005$.
[31] Aravind Srinivasan. Approximation algorithms for stochastic and risk-averse optimization. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1305-1313, 2007.
[32] Nicole Immorlica, David Karger, Maria Minkoff, and Vahab S. Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In Proc. 15th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 691-700, 2004.
[33] R. Ravi and Amitabh Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. In Integer Programming and Combinatorial Optimization, 2004.
[34] A. Gupta, R. Ravi, and A. Sinha. An edge in time saves nine: Lp rounding approximation algorithms for stochastic network design. In 45th Annual IEEE Symposium on Foundations of Computer Science, pages 218-227, 2004.
[35] Brian C. Dean, Michel X. Goemans, and Jan Vondrak. Approximating the stochastic knapsack problem: The benefit of adaptivity. Mathematics of Operations Research, pages 945-964, 2008.
[36] Rolf H. Mohring, Andreas S. Schulz, and Marc Uetz. Approximation in stochastic scheduling: The power of lp-based priority policies. J. ACM, pages 924-942, 1999.
[37] Brian Brubach, Darshan Chakrabarti, John P. Dickerson, Samir Khuller, Aravind Srinivasan, and Leonidas Tsepenekas. A pairwise fair and community-preserving approach to k-center clustering. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of Proceedings of Machine Learning Research, pages 1178-1189, 2020.
[38] Brian Brubach, Darshan Chakrabarti, John P. Dickerson, Aravind Srinivasan, and Leonidas Tsepenekas. Fairness, semi-supervised learning, and more: A general framework for clustering with stochastic pairwise constraints. Proceedings of the AAAI Conference on Artificial Intelligence, 35(8):6822-6830, May 2021.
[39] Jon Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. J.ACM, September 2002.
[40] Mikhail Bilenko, Sugato Basu, and Raymond J Mooney. Integrating constraints and metric learning in semi-supervised clustering. In Proceedings of the twenty-first international conference on Machine learning, page 11, 2004.
[41] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, pages 577-584, 2001.
[42] Jian Zhang and Rong Yan. On the value of pairwise constraints in classification and consistency. In Proceedings of the 24th International Conference on Machine Learning, 2007.
[43] Sugato Basu, Ian Davidson, and Kiri Wagstaff. Constrained Clustering: Advances in Algorithms, Theory, and Applications. Chapman \& Hall/CRC, 1 edition, 2008. ISBN 1584889969.
[44] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. Machine Learning, 109(2):373-440, 2020.
[45] Ian Davidson, S. S. Ravi, and Leonid Shamis. A sat-based framework for efficient constrained clustering. In Proceedings of the 2010 SIAM International Conference on Data Mining, 2010.
[46] Darshan Chakrabarti, John P. Dickerson, Seyed A. Esmaeili, Aravind Srinivasan, and Leonidas Tsepenekas. A new notion of individually fair clustering: $\alpha$-equitable $k$-center. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, Proceedings of The 25th International Conference on Artificial Intelligence and Statistics, volume 151 of Proceedings of Machine Learning Research, pages 6387-6408. PMLR, 28-30 Mar 2022.
[47] Dimitris Bertsimas, Vivek F. Farias, and Nikolaos Trichakis. The price of fairness. 59(1): 17-31, January 2011.
[48] Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, and Maria Kyropoulou. The efficiency of fair division. WINE '09, page 475-482, 2009.
[49] Samir Khuller and Yoram J. Sussmann. The capacitated k-center problem. SIAM Journal on Discrete Mathematics, 13(3):403-418, 2000.
[50] Nihesh Anderson, Suman K. Bera, Syamantak Das, and Yang Liu. Distributional individual fairness in clustering. CoRR, abs/2006.12589, 2020. URL https: //arxiv.org/ abs/2006.12589.
[51] Christopher Jung, Sampath Kannan, and Neil Lutz. A center in your neighborhood: Fairness in facility location, 2019.
[52] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In Advances in Neural Information Processing Systems 30, 2017.
[53] Michael Dinitz, Aravind Srinivasan, Leonidas Tsepenekas, and Anil Vullikanti. Fair disaster containment via graph-cut problems. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, Proceedings of The 25th International Conference on Artificial Intelligence and Statistics, volume 151 of Proceedings of Machine Learning Research, pages 6321-6333. PMLR, 28-30 Mar 2022.
[54] Harald Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In Cynthia Dwork, editor, Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008, pages 255264. ACM, 2008.
[55] Amy E. Babay, Michael Dinitz, Aravind Srinivasan, Leonidas Tsepenekas, and Anil Vullikanti. Controlling epidemic spread using probabilistic diffusion models on networks. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, Proceedings of The 25th International Conference on Artificial Intelligence and Statistics, volume 151 of Proceedings of Machine Learning Research, pages 11641-11654. PMLR, 28-30 Mar 2022.
[56] David R. Karger. Random sampling in cut, flow, and network design problems. Mathematics of Operations Research, pages 383-413, 1999.
[57] Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. SIAM Journal on Optimization, 12(2):479-502, 2002.
[58] Andrzej P Ruszczynski and Alexander Shapiro. Stochastic programming, volume 10. Elsevier, Amsterdam, 2003.
[59] Alexander Shapiro. Monte Carlo sampling methods. Handbooks in operations research and management science, 10:353-425, 2003.
[60] Chaitanya Swamy and David Shmoys. The sample average approximation method for 2-stage stochastic optimization. 042008.
[61] S. Eubank, V. S. Anil Kumar, M. V. Marathe, A. Srinivasan, and N. Wang. Structure of Social Contact Networks and Their Impact on Epidemics. In Discrete Methods in Epidemiology, volume 70, pages 179-200. American Math. Soc., Providence, RI, 2006.
[62] Brian Brubach, Nathaniel Grammel, David G. Harris, Aravind Srinivasan, Leonidas Tsepenekas, and Anil Vullikanti. Approximating Two-Stage Stochastic Supplier Problems. In Mary Wootters and Laura Sanità, editors, Approximation, Randomization, and

Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021), volume 207 of Leibniz International Proceedings in Informatics (LIPIcs), pages 23:1-23:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. ISBN 978-3-95977-207-5.
[63] Moses Charikar, Chandra Chekuri, and Martin Pal. Sampling bounds for stochastic optimization. In Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques, pages 257-269, 2005.
[64] Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. J. ACM, 1986.
[65] Lingxiao Huang, Shaofeng Jiang, and Nisheeth Vishnoi. Coresets for clustering with fairness constraints. In Advances in Neural Information Processing Systems 32, pages 75897600. Curran Associates, Inc., 2019.
[66] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. Scalable fair clustering. In Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 405-413, 2019.
[67] Ioana O. Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R. Schmidt, and Melanie Schmidt. On the Cost of Essentially Fair Clusterings. In APPROX/RANDOM 2019, volume 145, pages 18:1-18:22, 2019.
[68] Clemens Rösner and Melanie Schmidt. Privacy Preserving Clustering with Constraints. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, 45th International Colloquium on Automata, Languages, and Programming (ICALP 2018), volume 107 of Leibniz International Proceedings in Informatics (LIPIcs), pages 96:1-96:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-076-7.
[69] Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. Guarantees for spectral clustering with fairness constraints. In International Conference on Machine Learning, pages 3458-3467. PMLR, 2019.
[70] Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. Fair correlation clustering. In International Conference on Artificial Intelligence and Statistics, pages 4195-4205. PMLR, 2020.
[71] Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. In Advances in Neural Information Processing Systems 32, pages 4954-4965, 2019.
[72] 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, KDD '19, 2019.
[73] Sara Ahmadian, Alessandro Epasto, Marina Knittel, Ravi Kumar, Mohammad Mahdian, Benjamin Moseley, Philip Pham, Sergei Vassilvitskii, and Yuyan Wang. Fair hierarchical clustering. Advances in Neural Information Processing Systems, 33:21050-21060, 2020.
[74] Seyed A. Esmaeili, Brian Brubach, Leonidas Tsepenekas, and John P. Dickerson. Probabilistic fair clustering. In Neural Information Processing Systems (NeurIPS), 2020.
[75] Seyed A. Esmaeili, Brian Brubach, Aravind Srinivasan, and John P. Dickerson. Fair clustering under a bounded cost. In NeurIPS, 2021.
[76] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 3448-3457. PMLR, 09-15 Jun 2019.
[77] Matthew Jones, Huy Nguyen, and Thy Nguyen. Fair k-centers via maximum matching. In International Conference on Machine Learning, pages 4940-4949. PMLR, 2020.
[78] Suhas Thejaswi, Bruno Ordozgoiti, and Aristides Gionis. Diversity-aware k-median: Clustering with fair center representation. In Nuria Oliver, Fernando Pérez-Cruz, Stefan Kramer, Jesse Read, and Jose A. Lozano, editors, Machine Learning and Knowledge Discovery in Databases. Research Track, pages 765-780, Cham, 2021. Springer International Publishing. ISBN 978-3-030-86520-7.
[79] Xingyu Chen, Brandon Fain, Liang Lyu, and Kamesh Munagala. Proportionally fair clustering. In Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 1032-1041. PMLR, 09-15 Jun 2019.
[80] Evi Micha and Nisarg Shah. Proportionally Fair Clustering Revisited. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, 47th International Colloquium on Automata, Languages, and Programming (ICALP 2020), volume 168 of Leibniz International Proceedings in Informatics (LIPIcs), pages 85:1-85:16, Dagstuhl, Germany, 2020. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. ISBN 978-3-95977-138-2.
[81] Sayan Bandyapadhyay, Tanmay Inamdar, Shreyas Pai, and Kasturi Varadarajan. A Constant Approximation for Colorful k-Center. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, 27th Annual European Symposium on Algorithms (ESA 2019), volume 144 of Leibniz International Proceedings in Informatics (LIPIcs), pages 12:112:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-124-5.
[82] Xinrui Jia, Kshiteej Sheth, and Ola Svensson. Fair colorful k-center clustering. In IPCO, pages 209-222, 2020.
[83] Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. Fair clustering via equitable group representations. In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, page 504-514, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097.
[84] Mehrdad Ghadiri, Samira Samadi, and Santosh Vempala. Socially fair k-means clustering. In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, page 438-448, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097.
[85] Yury Makarychev and Ali Vakilian. Approximation algorithms for socially fair clustering. In Proceedings of Thirty Fourth Conference on Learning Theory, volume 134 of Proceedings of Machine Learning Research, pages 3246-3264. PMLR, 15-19 Aug 2021.
[86] Dishant Goyal and Ragesh Jaiswal. Tight fpt approximation for socially fair clustering, 2021.
[87] Mehrdad Ghadiri, Mohit Singh, and Santosh S. Vempala. Constant-factor approximation algorithms for socially fair $k$-clustering, 2022. URL https://arxiv.org/abs/ 2206.11210.
[88] Debajyoti Kar, Sourav Medya, Debmalya Mandal, Arlei Silva, Palash Dey, and Swagato Sanyal. Feature-based individual fairness in k-clustering, 2021.
[89] Sepideh Mahabadi and Ali Vakilian. Individual fairness for k-clustering. In Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research, pages 6586-6596. PMLR, 13-18 Jul 2020.
[90] Deeparnab Chakrabarty and Maryam Negahbani. Better algorithms for individually fair $k$-clustering, 2021.
[91] Ali Vakilian and Mustafa Yalçıner. Improved approximation algorithms for individually fair clustering, 2021.
[92] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. A notion of individual fairness for clustering, 2020.
[93] Saba Ahmadi, Pranjal Awasthi, Samir Khuller, Matthäus Kleindessner, Jamie Morgenstern, Pattara Sukprasert, and Ali Vakilian. Individual preference stability for clustering, 2022.
[94] Anupam Gupta, Martin Pal, R. Ravi, and Amitabh Sinha. Sampling and cost-sharing: Approximation algorithms for stochastic optimization problems. SIAM J. Comput., pages 1361-1401, 2011.
[95] Shipra Agrawal, Amin Saberi, and Yinyu Ye. Stochastic combinatorial optimization under probabilistic constraints, 2008.
[96] David G. Harris, Shi Li, Thomas Pensyl, Aravind Srinivasan, and Khoa Trinh. Approximation algorithms for stochastic clustering. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, pages 6041-6050, 2018.
[97] David G. Harris, Thomas W. Pensyl, Aravind Srinivasan, and Khoa Trinh. Dependent randomized rounding for clustering and partition systems with knapsack constraints. In The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, volume 108 of Proceedings of Machine Learning Research, pages 2273-2283, 2020.
[98] Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. Theoretical computer science, 38:293-306, 1985.
[99] Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. 13(2), 2017.
[100] Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing, STOC '02, page 731-740, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 1581134959.
[101] Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for $\$ \mathrm{k} \$$-means and euclidean $\$ \mathrm{k} \$$-median by primal-dual algorithms. SIAM Journal on Computing, 49(4):FOCS17-97-FOCS17-156, 2020.
[102] Euiwoong Lee, Melanie Schmidt, and John Wright. Improved and simplified inapproximability for k-means. Information Processing Letters, 120:40-43, 2017.
[103] Yingrui Yang, Ashley McKhann, Sixing Chen, Guy Harling, and Jukka-Pekka Onnela. Efficient vaccination strategies for epidemic control using network information. Epidemics, 27:115-122, 2019.
[104] Ken T.D. Eames, Jonathan M. Read, and W. John Edmunds. Epidemic prediction and control in weighted networks. Epidemics, 1(1):70-76, 2009.
[105] Reuven Cohen, Shlomo Havlin, and Daniel Ben-Avraham. Efficient immunization strategies for computer networks and populations. Phys. Rev. Lett., 91:247901, Dec 2003.
[106] Joel C. Miller and James Mac Hyman. Effective vaccination strategies for realistic social networks. pages 780-785, 2007.
[107] Prathyush Sambaturu, Bijaya Adhikari, B Aditya Prakash, Srinivasan Venkatramanan, and Anil Vullikanti. Designing effective and practical interventions to contain epidemics. In Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, pages 1187-1195, 2020.
[108] A. Ganesh, L. Massoulie, and D. Towsley. The effect of network topology on the spread of epidemics. Proceedings of INFOCOM, 2005.
[109] Victor M. Preciado, Michael Zargham, and David Sun. A convex framework to control spreading processes in directed networks. In Annual Conference on Information Sciences and Systems (CISS). IEEE, 2014.
[110] Victor M. Preciado, Michael Zargham, Chinwendu Enyioha, Ali Jadbabaie, and George J. Pappas. Optimal vaccine allocation to control epidemic outbreaks in arbitrary networks. In IEEE Conference on Decision and Control. IEEE, 2013.
[111] Victor M. Preciado, Michael Zargham, Chinwendu Enyioha, Ali Jadbabaie, and George J. Pappas. Optimal resource allocation for network protection against spreading processes. In IEEE Transactions on Control of Network Systems, pages 99 - 108. IEEE, 2014.
[112] Sudip Saha, Abhijin Adiga, B. Aditya Prakash, and Anil Vullikanti. Approximation algorithms for reducing the spectral radius to control epidemic spread. In SIAM SDM, 2015.
[113] Masaki Ogura and Victor M. Preciado. Optimal Containment of Epidemics in Temporal and Adaptive Networks, pages 241-266. Springer Singapore, Singapore, 2017.
[114] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http: //archive.ics.uci.edu/ml.
[115] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. Decision Support Systems, 62, 06 2014. doi: 10.1016/j.dss. 2014. 03.001.
[116] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, page 202-207. AAAI Press, 1996.
[117] Ivy Yeh and Che-Hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36:2473-2480, 03 2009. doi: 10.1016/j.eswa.2007.12.020.
[118] Cheng Tang and Claire Monteleoni. On lloyd's algorithm: New theoretical insights for clustering in practice. In Arthur Gretton and Christian C. Robert, editors, Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, volume 51 of Proceedings of Machine Learning Research, pages 1280-1289, Cadiz, Spain, 09-11 May 2016. PMLR.
[119] J. Mihelic and B. Robic. Solving the k-center problem efficiently with a dominating set algorithm. J. Comput. Inf. Technol., 13:225-234, 2005.
[120] Christopher Meek, Bo Thiesson, and David Heckerman. The learning-curve sampling method applied to model-based clustering. Journal of Machine Learning Research, 2: 397-418, 06 2002. doi: 10.1162/153244302760200678.
[121] Beata Strack, Jonathan Deshazo, Chris Gennings, Juan Luis Olmo Ortiz, Sebastian Ventura, Krzysztof Cios, and John Clore. Impact of hbalc measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records. BioMed research international, 2014:781670, 04 2014. doi: 10.1155/2014/781670.
[122] David P. Williamson and David B. Shmoys. The Design of Approximation Algorithms. Cambridge University Press, USA, 1st edition, 2011.
[123] Svante Janson. New versions of Suen's correlation inequality. Random Structures and Algorithms, 13(3-4):467-483, 1998.
[124] Irit Dinur and David Steurer. Analytical approach to parallel repetition. In Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, STOC '14, page 624-633, 2014.
[125] Howard Karloff. Linear Programming. Birkhauser Boston Inc., USA, 1991. ISBN 0817635610.
[126] M. Grötschel, L. Lovász, and A. Schrijver. Geometric Algorithms and Combinatorial Optimization. Springer, Berlin, Heidelberg, 1988.
[127] Uriel Feige. On sums of independent random variables with unbounded variance and estimating the average degree in a graph. SIAM Journal on Computing, 35(4):964-984, 2006.
[128] Lap-Chi Lau, R. Ravi, and Mohit Singh. Iterative Methods in Combinatorial Optimization. Cambridge University Press, USA, 1st edition, 2011. ISBN 0521189438.
[129] Deeparnab Chakrabarty and Maryam Negahbani. Generalized center problems with outliers. ACM Trans. Algorithms, 2019.
[130] Andrea Pietracaprina, Geppino Pucci, and Federico Soldà. Coreset-based strategies for robust center-type problems, 2020. URL https://arxiv.org/abs/2002.07463.
[131] Lydia T. Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pages 6196-6200. International Joint Conferences on Artificial Intelligence Organization, 72019.
[132] Omer Ben-Porat, Fedor Sandomirskiy, and Moshe Tennenholtz. Protecting the protected group: Circumventing harmful fairness. Proceedings of the AAAI Conference on Artificial Intelligence, 35(6):5176-5184, May 2021.


[^0]:    ${ }^{1}$ A distance function satisfies the following properties. i) $\left.\mathrm{d}(\mathrm{x}, \mathrm{x})=0, \mathbf{i i}\right) \mathrm{d}(\mathrm{x}, \mathrm{y})=\mathrm{d}(\mathrm{y}, \mathrm{x})$ and 3) for every $x, y, z$ we have $d(x, z) \leq d(x, y)+d(y, z)$. The last property is known as the triangle inequality.

[^1]:    ${ }^{2}$ For a vector $\alpha=\left(\alpha_{1}, \alpha_{2}, \ldots, \alpha_{k}\right)$ and a subset $X \subseteq\{1,2, \ldots, k\}$, we use $\alpha(X)$ to denote $\sum_{i \in X} \alpha_{i}$

[^2]:    ${ }^{3}$ We use $[k]$ to denote $\{1, \ldots, k\}$ for some integer $k \geq 1$

[^3]:    4"Paths" will refer throughout to simple paths: ones in which no nodes or edges are repeated.

[^4]:    ${ }^{1}$ In these plots, for the two baseline algorithms we excluded points with $f_{j}^{P P}=+\infty$ or $f_{j}^{A G}=+\infty$ in the computation of $\max _{j} f_{j}^{P P}$ and $\max _{j} f_{j}^{A G}$. In other words, we were very lenient with the two baselines.

[^5]:    ${ }^{1} S(D, k)$ can be interpreted as the set of all vectors $x \in \mathbb{Z}_{\geq 0}^{D-w_{\min }+1}$ with $\ell^{1}$ norm equal to $k$. Our notation can be thought of as re-indexing $x$, so that the numbering starts at $w_{\text {min }}$ instead of 1 , and then finishes at $D$.

