University of Wrocław
Faculty of Mathematics and Computer Science

Krzysztof Sornat

Approximation Algorithms for Multiwinner Elections and Clustering Problems

PhD thesis

Supervisor
dr hab. Jarosław Byrka
Institute of Computer Science
University of Wrocław

June 2019

to my wife Kasia

Acknowledgements

First and foremost, I would like to thank my advisor, Jarosław Byrka. He guided me with patience and supported me for many years of my doctoral studies at the University of Wrocław. Before, he was my lecturer who introduced me to the theory of linear programming and algorithmic game theory. This was the beginning of my research interests. I also want to thank Jarek for many hours of discussions and showing me how to conduct research in a positive and sustainable manner. I appreciate a lot of freedom Jarek gave me in choosing my research activities. I am grateful to Jarek for introducing me to many researchers, with whom I was working later. Finally, I thank Jarek for many hours spent on the climbing wall Tarnogaj. Sometimes I needed to look at the things from some distance (for instance, from 14 meters above the floor).

I would like to give my special thanks to Piotr Faliszewski for our first meeting during FIT 2016 in Warsaw. Then Piotr successfully convinced me to do more research on approximation algorithms in computational social choice problems. Indeed, Piotr had strong impact on my research interests. I would like to thank Piotr for many hours of inspiring discussions and all exchanged e-mails.

I am particularly grateful to Piotr Skowron for many discussions on computational social choice and clustering problems. Our common paper is one of the pillars of this thesis. I thank Piotr for hosting me in TU Berlin for a one-week visit. His experience on working in academia he shared with me has an impact on my important decisions.

I thank Joachim Spoerhase for many hours of hard working together at blackboards in Würzburg and Wrocław. He taught me a lot about clustering problems and dependent LP-rounding techniques. I admire his ability to explain difficult ideas in simple words. I also thank Joachim for hosting me at the University of Würzburg for a two-week visit and the nice time spent together on Würzburg sightseeing.

My sincere thanks also goes to to Łukasz Kowalik for his positive attitude and supervision during my one-semester doctoral internship at the University of Warsaw. We run together the 11th Warsaw Halfmarathon however Łukasz was faster than me by 7 seconds. I am still training my physical fitness and trying to beat him.

I thank Marek Cygan for a successful collaboration and his assistance in solving my funding issues.

I would like to thank Evangelos Markakis for research cooperation and his supervision during my 3-month doctoral internship at Athens University of Economics and Business, Greece. I appreciate the time spent on a hiking trip around Dirfi mountain with his friends.

I am grateful to Fabrizio Grandoni for an opportunity of doing research in his

group at IDSIA, Lugano, Switzerland. I would like to thank Fabrizio for introducing me to survivable network desing problems and fruitful collaboration. I really enjoyed the 6-month stay in Lugano not only because of research but also as a beautiful place with an amazing landscape.

Besides the persons mentioned above, I was fortunate to work with many inspiring people. I have learned a lot from them. Especially I would like to thank all my great co-authors not mentioned above for a successful collaboration: Georgios Amanatidis, Arkadiusz Socała, Peter Fulla, Pasin Manurangsi, Waldo Gálvez, Afrouz Jabalameli.

I thank all my roommates and colleagues. Especially I would like to thank Mateusz Lewandowski for many hours of discussions about everything. Also, I am really grateful to Bartosz Rybicki who introduced me to doctoral studies and shared his experience. His hard-working motivates me to do the same. I would like to thank Andrzej Grzesik, who I met at the University of Warsaw, for talks about working in academia which were important for my future desicions made.

I thank all institutions that funded my research. Especially, I was supported by the National Science Centre, Poland, grant numbers 2015/17/N/ST6/03684 and 2018/28/T/ST6/00366, the Swiss National Science Foundation (SNSF) Grant APX-NET 200021_159697/1 and the Foundation for Polish Science (FNP) START scholarship.

Last, and most important, I would like to acknowledge the constant support of my family. Especially, I thank my wife Kasia for her love and lifting my spirit when I needed it most. I admire her patience and hopefulness during all of the years of my doctoral studies.

Abstract

The thesis is devoted to polynomial time approximation algorithms for a few NP-hard discrete optimization problems that model real-world issues such as clustering and multiwinner elections.

Multiwinner elections and clustering problems have very similar settings. Our goal in both is to choose a fixed number of objects among a finite set of alternatives. We want to optimize the function of distances/disagreement between input points/voters and cluster centers/winners.

We exploit similarities and explore the differences in order to understand when efficient algorithms in one model can be useful in another. The main contributions of the thesis are the following algorithms and proofs of their approximation factors:

1. The first polynomial time constant-factor approximation algorithm for the Ordered k-Median problem. The constant achieved is equal to $38 + \epsilon$ for any $\epsilon > 0$. This improves the previous best $\mathcal{O}(\log n)$-approximation achieved by Aouad and Segev [7].
2. The first polynomial time constant-factor approximation algorithm for the Rectangular Ordered k-Median problem. The constant achieved is equal to 15. The previous best approximation was $\mathcal{O}(\log n)$ [7].
3. The first polynomial time constant-factor approximation for the Harmonic k-Median problem in general spaces (not only metric). The constant is upperbounded by 2.36. To the best of our knowledge this is the first natural variant of k-Median that has constant-factor approximation without assuming the triangle inequality.
4. The first PTAS (polynomial time approximation scheme) for the Minimax Approval Voting problem. This improves the previous best 2-approximation [39].
5. A parameterized approximation scheme for Minimax Approval Voting parameterized by the value d of an optimal solution. The running time is upperbounded by $(3/\epsilon)^{2d}(nm)^{\mathcal{O}(1)}$. It is essentially optimal assuming the Exponential Time Hypothesis due to Cygan et al. [56]. The parameterized approximation scheme allows us to construct a faster PTAS for Minimax Approval Voting.

We believe that the ideas and algorithm analysis techniques developed in this thesis will be useful in further work on approximation algorithms. We also hope our results will stimulate more interdisciplinary research on relations between clustering problems and multiwinner elections.

Keywords: approximation algorithms, linear programming, LP-rounding, parameterized algorithms, clustering, k-median, facility location, multiwinner election, proportional approval voting, minimax approval voting, computational social choice

Streszczenie

Rozprawa doktorska jest poświęcona algorytmom aproksymacyjnym rozwiązującym wybrane NP-trudne problemy optymalizacji dyskretnej w czasie wielomianowym.

Rozważane problemy obliczeniowe modelują zagadnienia znane z przemysłowych i społecznych zastosowań: klastrowanie oraz wybór wielu zwycięzców. W obu rodzajach problemów naszym celem jest wybranie określonej liczby obiektów spośród skończonego zbioru alternatyw; przy tym wyborze kierujemy się optymalizowaniem pewnej funkcji odległości (niezadowolenia) pomiędzy punktami danych (głosującymi), a centrami klastrów (zwycięzcami).

W trakcie pracy nad nowymi algorytmami rozważaliśmy powyższe podobieństwa, jak i różnice pozwalające lepiej zrozumieć, kiedy algorytmy efektywne dla jednego zagadnienia mogą być użyteczne także w drugim. Głównym wkładem tej rozprawy doktorskiej jest konstrukcja poniżej wymienionych algorytmów oraz analiza ich współczynników aproksymacji:

1. Pierwszy algorytm o stałym współczynniku aproksymacji dla problemu Ordered k-Median. Osiągnięta stała wynosi $38 + \epsilon$ dla dowolnego $\epsilon > 0$. W ten sposób poprawiliśmy poprzedni najlepszy współczynnik aproksymacji, który wynosił $\mathcal{O}(\log n)$, a został osiągnięty przez Aouada oraz Segeva [7].

2. Pierwszy algorytm o stałym współczynniku aproksymacji dla problemu Rectangular Ordered k-Median. Osiągnięta stała wynosi 15. Poprzedni najlepszy współczynnik aproksymacji wynosił $\mathcal{O}(\log n)$ [7].

3. Pierwszy algorytm o stałym współczynniku aproksymacji dla problemu Harmonic k-Median w ogólnych przestrzeniach (nie tylko metrycznych). Osiągnięta stała jest ograniczona z góry przez 2.36. Zgodnie z naszą wiedzą jest to pierwszy naturalny wariant problemu k-Median, który potrafimy aproksymować ze stałym współczynnikiem bez zakładania nierówności trójkąta.

4. Pierwszy schemat aproksymacji wielomianowej (PTAS) dla problemu Minimax Approval Voting. Poprzedni najlepszy współczynnik aproksymacji osiągnięty przez Caragiannisa i innych autorów [39] wynosił 2.

5. Schemat aproksymacji w czasie parametryzowanym dla problemu Minimax Approval Voting parametryzowanego wartością optymalnego rozwiązania, którą oznaczamy d. Czas działania jest ograniczony z góry przez $(3/\epsilon)^{2d}(nm)^{\mathcal{O}(1)}$. Cygan i inni autorzy [56] pokazali, że zasadniczo jest to najszybszy tego typu

algorytm przy założeniu Hipotezy Czasu Wykładniczego. Skonstruowany algorytm pomógł nam w opracowaniu szybszego schematu aproksymacji wielomianowej dla Minimax Approval Voting.

Wierzymy, że użyte pomysły i techniki analizy algorytmów zawarte w tej rozprawie będą użyteczne w dalszych pracach nad algorytmami aproksymacyjnymi. Mamy również nadzieję, że nasze wyniki będą stymulowały następne badania nad relacjami pomiędzy problemami klastrowania oraz wyborami wielu zwycięzców.

Słowa kluczowe: algorytmy aproksymacyjne, programowanie liniowe, zaokrąglanie rozwiązań programów liniowych, algorytmy parametryzowane, klastrowanie, k-median, umiejscawianie fabryk, wybory wielu zwycięzców, proportional approval voting, minimax approval voting, obliczeniowa teoria wyboru społecznego

Contents

Chapter 1

Introduction

"All models are wrong but some are useful". This sentence was stated by British statistician George Box in the late '70s [24]. There is a lot of motivation for research on mathematical modelling because of practical needs. Unfortunately, modelling complex systems can be too difficult and therefore often we do so by using only a few mathematical formulas and making strong assumptions. Mathematical models are often just a simplification and idealization of a system or natural phenomena that one wants to understand and simulate. As a consequence, it seems to be clear that all models are wrong as they do not describe the complexity of a problem exactly as it is in reality. On the other hand, depending on the purpose of the modelling, a model can be less or more accurate, i.e., useful.

Mathematical optimization is useful in modelling different issues in economics, mechanics, civil engineering, operations research, machine learning, etc. Choosing a proper optimization problem that models a real-world optimization issue is a crucial step. Once the problem is chosen, there are two primarily questions one can ask: how to find an optimal solution and how effectively (quickly) can it be done? Here computer science can provide some help. Computer science is not about choosing a proper model and making fair assumptions, but it focuses on effective solving and understanding the structure of computational problems.

Computing an optimal solution may need a lot of computational resources (time, memory, etc.). Computational complexity theory (a subfield of computer science) studies the hardness of finding solutions to computational problems. For example, it has been proven that many fundamental optimization problems like the Maximum Satisfiability Problem, the Travelling Salesman Problem and the Maximum Clique Problem are NP-hard [51, 93, 105], i.e., they are at least as hard as any problem contained in a complexity class called NP (Nondeterministic Polynomial time) [8]. NP-hard problems cannot be solved in time polynomial in the input size, assuming $\mathsf{P} \neq \mathsf{NP}$. Indeed, it is widely believed that $\mathsf{P} \neq \mathsf{NP}$ [76]. The "P versus NP" problem stated by Stephen Cook [51] seems to be the most important open problem in computer science. It is one of the Millennium Prize Problems listed by Clay Mathematics Institute in 2000 [40]. The institute funds a prize of 1 milion US dollars for the discoverer(s). For more about computational complexity theory see a classical book of Arora and Barak [8].

In recent years researchers in computational complexity theory intensively examine the answers to the questions: how good solution can we find using limited computational resources? There is a tradeoff between the quality of the solution and the running time of finding it. We say that an algorithm $\mathcal{A}$ is *an α-approximation algorithm* for a minimization problem $\mathcal{P}$ if for any instance $\mathcal{I}$ of $\mathcal{P}$ we have

$$\text{cost}(\mathcal{A}(\mathcal{I})) \leq \alpha \cdot \text{OPT}(\mathcal{I}),$$

where $\text{cost}(\cdot)$ is an objective function of $\mathcal{P}$, $\mathcal{A}(\mathcal{I})$ is a solution returned by an algorithm $\mathcal{A}$ and $\text{OPT}(\mathcal{I})$ is the value of an optimal solution to an instance $\mathcal{I}$. We are focused on polynomial time approximation algorithms (unless we state otherwise) as polynomial time algorithms are seen to be efficient.

α-approximation algorithm gives a solution that is at most α times worse than an optimal solution. α is called *the approximation ratio* or *the approximation factor* and in general, α can be a function of an instance's parameters. The closer α to 1, the better. Achieving an approximation ratio equal to a constant is often a primarly goal because then the quality of a solution is independent of the input size. The secondary goal is to have the constant as small as possible. Ideally if we can get *a polynomial time approximation scheme* (PTAS) that is an algorithm which takes any $\epsilon > 0$ (as an additional input) and returns $(1+\epsilon)$-approximate solution in time polynomial in the input size. Note that there can be any dependence on ϵ in the running time, i.e., it can be exponential in $1/\epsilon$. For more about approximation algorithms see for example the book by Williamson and Shmoys [148].

The main issue considered in this thesis is computing in polynomial time an approximation of an optimal solution to a few NP-hard discrete optimization problems that model real-world issues such as clustering and multiwinner elections.

Clustering

Clustering a given set of objects into k groups that display a certain internal proximity is a profound combinatorial optimization setting. In a typical setup, we represent the objects as points in a metric space and evaluate the quality of the clustering by a certain function of distances within the clusters. There are many objective functions and assumptions considered dependent on the application. In a centroid based model of clustering, every cluster has a center. Probably the best known centroid clustering problem is the k-Means problem [85, 141] which objective function is to minimize the sum of squared distances from objects to their cluster centers. If the objective is to minimize the sum of the distances to a cluster center, we call the resulting optimization problem k-Median [46, 79, 80, 91]. If the objective is to minimize the maximal distance to a cluster center, then we talk about the k-Center problem [79, 82].

Facility Location

The setting of clustering is similar to *facility location* problems [14, 72] which have as a goal opening facilities (centers) that serve the clients (demands). Each possible

facility location has price for opening a facility. It can model the cost of building a service center. After opening a set of facilities, a client is served by their closest open facility and the cost of service is equal to the distance between them (cost of transport can be modeled as a linear function of distance). The objective function in the Uncapacitated Facility Location problem [14, 53, 135] is to minimize the sum of total connection (service) cost and total opening cost. Adding one more assumption, i.e., *capacities* of the facilities (the number of clients that can be served simultanously by a particular facility) we get a problem called Capacitated Facility Location (CFL) [123]. We can use CFL to model locating hospitals, schools or warehouses which have limited capacity.

Multiwinner Elections

Clustering problems such as k-Median, k-Means or k-Center can be seen as multiwinner elections in which we want to choose exactly k winners (cluster centers) from a set of candidates. In many multiwinner election rules, our goal is to minimize the function of dissatisfaction (corresponding to distance) over voters (corresponding to data points). Indeed, k-Median problem corresponds to a minimization variant of the well known Chamberlin-Courant voting rule [45] in which we care only about utility to its closest winning candidate.

Approximation algorithms for many election rules have been extensively studied in the literature. In the world of single-winner rules, there are already very good approximation algorithms known for the Kemeny rule [2, 52, 96] which minimizes the sum of the Kendall's tau distances[1] [94, 95] and for the Dodgson rule [37, 38, 68, 83, 114] which minimizes the number of swaps needed to achieve a Condorcet winner, i.e., a candidate that wins all pairwise comparisons with all the other candidates [21, 59]. A hardness of approximation has been proven for the Young rule [37]. For the multiwinner case we know good (randomized) approximation algorithms for the Chamberlin-Courant rule [137], the Monroe rule [137], or the maximization variant of Proportional Approval Voting [136].

Using approximation and randomized algorithms for finding winners of elections requires some comment because their outcome can be non-optimal and additionally it can be different for different random bits. While using approximation or randomization in domains similar to political elections may appear controversial, multiwinner elections have much more diverse applications—such applications include aggregating preferences of individual agents [50], finding a set of results a search engine should display [64], recommending a set of products a company should offer to its customers [111, 112], allocating shared resources among agents [117, 136], solving variants of segmentation problems [98], or even improving genetic algorithms [70].

However, even for political elections, the use of approximation algorithms is a promising direction. One approach is to view an approximation algorithm as a new, full-fledged voting rule. Indeed, Sequential Proportional Approval Voting [147], that was used briefly in Sweden during the early 1900s [12], is a greedy

[1] Also called "bubble sort" distance as it measures similarity of two ranking lists by the number of pairwise disagreements.

algorithm that gives an approximate solution to Proportional Approval Voting. For more discussion and examples of approximation algorithms as full-fledged voting rules we refer to the following papers [12, 37, 38, 65, 137].

The use of randomized algorithms in multiwinner elections has been advocated in the literature as well—e.g., one can arrange an election where each participant is allowed to suggest a winning committee, and the best out of the suggested committees is selected; in such case the approximation guaranty of the algorithm corresponds to the quality of the outcome of elections. For a more detailed discussion see the work of Skowron et al. [137].

Finally, randomized and approximation algorithms seem to be well justified for high-frequency decisions [11], e.g., online scheduling or online systems.

Relations Between Multiwinner Elections and Clustering Problems

The same objects are given different names in different research areas such as: operations research, artificial intelligence or social choice theory. Computational issues in the just mentioned research areas formed new interdisciplinary research topics, i.e., combinatorial optimization [134], machine learning [4] and computational social choice [28, 66] respectively. In Table 1.1 we systematize terminology mentioned previously and which will be used interchangeably in this thesis. The purpose of Table 1.1 is to highlight problems considered in this thesis that come from different research fields but have similar settings. Some entries are debatable, for example many voting rules for multiwinner elections do not define representants of particular candidates. The table is not complete and obviously there are some simplifications. For example, in clustering and facility location problems it is almost always assumed that objects lie in a metric space. This assumption is too strong for multiwinner elections, although often instances are restricted in other ways, for example, every voter defines the utility of choosing particular candidates assigning rates from a fixed set of rates. In the Borda rule [150] a voter define bijection between set of rates $\{m-1, m-2, \ldots, 1, 0\}$ and all the candidates[2]. The space defined is not necessarily metric but has another useful structure.

In the discussion above, we do not claim that the models from different fields are the same. In fact they reflect different purposes of introducing these models. We are going to take advantage of the similarities of the models to provide solutions used successfully in different areas. From the other side, the differences are explored in order to understand why efficient solutions in one model do not work well for another.

[2]In fact the rates in the Borda rule follows directly from the orders over candidates. Each voter gives his or her preference order over all candidates. Each candidate gets as many points as many candidates are below in the preference orders. This means the top candidate wins with $m-1$ other candidates, the second top candidate wins with $m-2$ other candidates, etc.

	combinatorial optimization	**machine learning**	**computational social choice**
problem name	facility location	(centroid) clustering	multiwinner elections
object to be chosen	facility	potential cluster center	candidate/item
input object	client/demand	data point	voter/agent
measure	connection cost	distance function	dissatisfaction/utility
object chosen	open facility	cluster center	winner
group of similar objects	clients served by the same facility	cluster	voters represented by the same winner
size upper bound	capacity	max. size of a cluster	max. number of represented voters

Table 1.1: Objects considered in this thesis have different names in different research fields. Each column contains terminology used in a particular research area.

1.1 Main Contributions

The main contributions of the thesis are the following algorithms and proofs of their approximation factors.

1. The first polynomial time constant-factor approximation algorithm for the Ordered k-Median problem. The constant achieved is equal to $38 + \epsilon$ for any $\epsilon > 0$. This improves the previous best $\mathcal{O}(\log n)$-approximation achieved by Aouad and Segev [7].

2. The first polynomial time constant-factor approximation algorithm for the Rectangular Ordered k-Median problem. The constant achieved is equal to 15. The previous best approximation was $\mathcal{O}(\log n)$ [7].

3. The first polynomial time constant-factor approximation for the Harmonic k-Median problem in general spaces (not only metric). The constant is upperbounded by 2.36. To the best of our knowledge this is the first natural variant of k-Median that has constant-factor approximation without assuming the triangle inequality.

4. The first PTAS (polynomial time approximation scheme) for the Minimax Approval Voting problem. This improves the previous best 2-approximation [39].

5. A parameterized approximation scheme for Minimax Approval Voting parameterized by the value d of an optimal solution. The running time is upperbounded by $(3/\epsilon)^{2d}(nm)^{\mathcal{O}(1)}$. It is essentially optimal assuming the Exponential Time Hypothesis due to Cygan et al. [56]. The parameterized approximation scheme allows us to construct a faster PTAS for Minimax Approval Voting.

Developed algorithmic techniques might be of independent interest as they can be useful in further work on approximation algorithms. Also we hope our results will stimulate more interdisciplinary research on relations between clustering problems and multiwinner elections.

1.2 Outline

This thesis is organized as follows. In Section 1.3, we introduce the problems considered in the thesis. In each subsection, we formally define problems, give the motivation, describe related works and state the main theorems of the thesis. We also summarize the ideas that led us to our solutions and give intuitions behind our algorithms. This should be enough for understanding the main problems solved, techniques used and overall contribution. Formal and detailed proofs for each of the theorems are contained in Chapters 2-4. In Section 1.4, we point out research papers in which the results of the thesis have appeared. In Chapter 5, we conclude the thesis by final remarks and we state open questions for further work.

1.3 Problems Considered

In this section we define problems formally, give motivation for studying them, point out related works and state the main results of the thesis. For full proofs we refer to Chapters 2-4 but here we summarize the main ideas used in the proofs.

1.3.1 Ordered k-Median

k-Median and k-Center[3] are two approaches to clustering that represent two extrems in their dependence on the variance between the indiviual connection costs in the evaluated solution. They are defined as follows.

k-Median
Input:
$\mathcal{F}$: a set of facilities, $|\mathcal{F}| = m$,
$\mathcal{C}$: a set of clients, $|\mathcal{C}| = n$,
k: a positive integer as the number of facilities to open,
c: a metric cost function, $c\colon (\mathcal{F} \cup \mathcal{C}) \times (\mathcal{F} \cup \mathcal{C}) \to \mathbb{R}_{\geq 0}$.
Notation:
$c_j(\mathcal{W}) := \min_{i \in \mathcal{W}} c_{ij}$, for $\mathcal{W} \subseteq \mathcal{F}, j \in \mathcal{C}$, is the smallest *connection cost of* j to a facility in $\mathcal{W}$.
Output: A set $\mathcal{W} \subseteq \mathcal{F}, |\mathcal{W}| = k$ that minimizes *the sum of connection costs*:
$$\sum_{j \in \mathcal{C}} c_j(\mathcal{W}).$$

[3]We note that k-Center is often defined as a clustering problem, i.e., $\mathcal{F} = \mathcal{C}$ is assumed [82]. k-Supplier is a more general problem in which $\mathcal{F}$ and $\mathcal{C}$ can be different [148].

k-Supplier (k-Center)
Input: The same as in k-Median.
Output: A set $\mathcal{W} \subseteq \mathcal{F}, |\mathcal{W}| = k$ that minimizes *the maximal connection cost*:

$$\max_{j \in \mathcal{C}} c_j(\mathcal{W}).$$

We study a generalization of both k-Median and k-Center, called the Ordered k-Median problem, where the connection costs are sorted non-increasingly and a non-increasing weight vector is applied to flexibly penalize the desired fraction of the highest costs.

Ordered k-Median
Input: The same as in k-Median, and additionally
$w = (w_1, \ldots, w_n)$: a non-increasing weight vector, $w_j \geq 0$.
Notation:
$c^{\rightarrow}(\mathcal{W}) = (c_j^{\rightarrow}(\mathcal{W}) : 1 \leq j \leq n)$ is a non-increasing vector of elements from $\{c_j(\mathcal{W}) : j \in \mathcal{C}\}$, where $\mathcal{W} \subseteq \mathcal{F}$.
Output: A set $\mathcal{W} \subseteq \mathcal{F}, |\mathcal{W}| = k$ that minimizes *the total connection cost*:

$$\sum_{j=1}^{n} w_j \cdot c_j^{\rightarrow}(\mathcal{W}).$$

We consider metric cost functions because the non-metric cost function does not allow us to obtain any non-trivial approximation (unless $\mathsf{P} = \mathsf{NP}$).

Works Related to Ordered k-Median.

The Ordered k-Median problem generalizes many fundamental clustering and location problems [9, 31, 46, 82, 86, 107, 145, 146] such as k-Median, k-Center problems, the k-Centdian problem where the objective is a convex combination of the k-Median and the k-Center objective, or the k-Facility ℓ-Centrum problem where the objective accounts for the ℓ highest connection costs. The ordered median objective function has also been considered in robust optimization [17, 18, 19, 129] and multi-objective optimization [73]. For a comprehensive overview we refer the reader to the books [101, 121] on Ordered k-Median problems and to dedicated works [25, 119, 120, 124].

The generality of Ordered k-Median renders it intriguing from the computational perspective [7]. For example, whereas k-Median and k-Center can be solved efficiently on trees by dynamic programming, such approaches seem to fail for Ordered k-Median due to the lack of separability properties [130]. Regarding approximability in general metric spaces, constant-factor approximation algorithms are long known for k-Median [46] and k-Center [82]. In contrast, even developing constant-factor approximations for Ordered k-Median with seemingly simple topologies such as trees turned out to be non-trivial [7]. In particular, due to the non-linearity of the objective function there seems to be no obvious way to apply tools such as metric tree embeddings [7, 15]. Not even non-trivial *super-constant* approximability results were known for Ordered k-Median until very recently, when

Aouad and Segev [7] were able to devise an $\mathcal{O}(\log n)$ approximation algorithm for the problem using a sophisticated local-search approach and the concept of *surrogate models*.

To demonstrate the highly non-local and dependent structure of the objective function, note that even if the clusters are given, the selection of the cluster centers cannot be made solely on a per-cluster basis but depends on the decision in other clusters due to the ranking of distances in the objective.

Due to the above-outlined difficulties in obtaining algorithmic results for the general problem, structural properties of continuous network spaces have been studied [61, 67, 127, 132] where facilities may be placed at interior points on edges, single-facility models [62, 67, 89, 120], and multi-facility models on special topologies such as trees [88, 129, 145]. Furthermore, integer programming formulations [119], branch-and-bound methods [22, 126], heuristics [60, 100, 125, 140], and related location models [128, 130] have been studied. For a survey on the topic we refer the reader to [121].

Below, we list approximability results for certain specific objective functions that fall under the framework of Ordered k-Median or closely related and where approximability has been studied for general metrics.

Approximation Algorithms for k-Median, k-Center, and k-Means.

k-Median admits constant-factor approximations via local-search [9], or a direct rounding of the standard LP [48]. The current best ratio of $(2.675+\epsilon)$ [31] is obtained by combining a primal-dual algorithm [86], and a nontrivial rounding of a so-called bi-point solution based on preprocessing introduced in [107].

The situation with the k-Center setting is simpler. A simple 2-approximation is obtained via guessing the longest connection distance in the optimal solution [82], and this is tight assuming $\mathsf{P} \neq \mathsf{NP}$ (the result holds when $\mathcal{F} = \mathcal{C}$). In the setting $\mathcal{F} \neq \mathcal{C}$ (called k-Supplier) the Hochbaum and Shmoys method [82] gives a tight 3-approximation [148]. Notably, by contrast to the k-Median setting, the most natural LP for k-Center has unbounded integrality gap.

Also k-Means admits a constant-factor approximation. The $(9+\epsilon)$-approximation local search algorithm for Euclidean k-Means [90] can be shown to provide 25-approximation in general metrics. The recent work of Ahmadian et al. [1] decreases these ratios to 6.357 and $9 + \epsilon$, respectively.

Approximation Algorithms for Further Specific Objective Functions.

A special case of Ordered k-Median that we call the Rectangular Ordered k-Median problem was considered by Tamir [145] (who called it k-Facility ℓ-Centrum). In this setting, we have to open exactly k facilities and the objective function is just a sum of ℓ largest client connection costs. We state the problem formally as follows.

RECTANGULAR ORDERED k-MEDIAN (k-FACILITY ℓ-CENTRUM)
Input: The same as in ORDERED k-MEDIAN with the following restriction: $w_i = 1$ for $i \leq \ell$ and $w_i = 0$ otherwise, for some $\ell \in \{1, 2, \ldots, n\}$.
Output: A set $\mathcal{W} \subseteq \mathcal{F}, |\mathcal{W}| = k$ that minimizes *the total connection cost*:

$$\sum_{j=1}^{\ell} c_j^{\rightarrow}(\mathcal{W}).$$

Note that RECTANGULAR ORDERED k-MEDIAN with $\ell = 1$ and $\ell = n$ is equivalent to k-CENTER and k-MEDIAN respectively. Tamir [145] gives polynomial time algorithms that solve the problem (optimally) on path and tree graphs. Obtaining a constant-factor approximation for RECTANGULAR ORDERED k-MEDIAN, however, has been an open problem [7, 145].

One should also notice at least two further, very recent works combining the k-CENTER objective and the k-MEDIAN objective. First, Alamdari and Shmoys [3] considered a bicriteria approximation algorithm for the k-CENTER and k-MEDIAN problems, i.e., the objective function is a linear combination of two objectives that are maximal connection cost in use and sum of all used connection costs. This problem is known as k-CENTDIAN [146]. They obtained polynomial time bicriteria approximation of $(4, 8)$, where the first factor is in respect to the k-CENTER objective and the second factor is in respect to the k-MEDIAN objective. (Alamdari and Shmoys note, however, that the two problems k-MEDIAN and k-CENTER are *not* approximable simultaneously.) Also k-CENTDIAN is a special case of ORDERED k-MEDIAN. The second recent work combining the k-MEDIAN and the k-CENTER objective is the work of Haris et al. [81] who propose a method to select k facilities that deterministically guarantees each client to have a connection within a certain fixed radius but also provides a stronger per client bound on cost expectation.

Our Results and Techniques

Our main result is the first constant-factor approximation algorithm for the ORDERED k-MEDIAN problem (Theorem 16) which improves the previous best $\mathcal{O}(\log n)$-approximation [7].

Theorem 16. *For any $\epsilon > 0$, there exists a randomized algorithm for* ORDERED k-MEDIAN *that computes expected $(38 + \epsilon)$-approximate solution in polynomial time (specifically $(nm)^{\mathcal{O}(1/\epsilon \log(1/\epsilon))}$).*

We are not aware of an LP relaxation for ORDERED k-MEDIAN with bounded integrality gap. In our approach we guess a *reduced cost function* roughly mimicking the weighting of distances in an optimum solution and solve the natural LP relaxation for k-MEDIAN under this reduced cost function (rather than under the original metric). Subsequently, we round this solution via a dependent LP rounding process by Charikar and Li [48] for k-MEDIAN operating on the original (unweighted) metrics.

The challenge and our main technical contribution consists in analyzing the approximation performance of this approach. In the original analysis of Charikar and Li [48] for the k-MEDIAN objective, a per-client bound on the expected connection cost

of this client with respect to its fractional connection cost is established. The global approximation ratio is then obtained by linearity of expectation. The above-described non-linear, ranking-based character of the objective of Ordered k-Median poses an obstacle to apply an analogous reasoning also in our more general setting as the actual weight that is applied to the connection cost of a client depends highly on the (random) connection costs of the other clients.

We use four key ingredients to overcome this technical hurdle.

First, we show in Theorem 5 that the algorithm provides a constant-factor approximation for rectangular weight vectors.

Theorem 5. *There exists a randomized algorithm for* Rectangular Ordered k-Median *that computes expected* 15*-approximate solution in polynomial time.*

This already answers the open problem stated in [3, 7]. In our analysis, the connection cost of a single client is partly charged to a *deterministic budget* related to a *combinatorial bound* based on guessing, and partly to a *probabilistic budget* whose expected value is bounded with respect to the fractional LP-solution. This approach allows us to limit the above-described problematic effect of the variance of individual client connection costs on the value of the ordered objective function of Ordered k-Median.

Second, we show a surprising *modularity* of Charikar and Li's rounding process (Lemma 11). The solution computed by this process can be related to the above-mentioned combinatorial and fractional bounds simultaneously with respect to *all* rectangular objectives. This property is oblivious to the objective with respect to which the input fractional solution was optimized.

Third, we *decompose* an arbitrary non-increasing weight vector into a convex combination of rectangular objectives. The aforementioned modularity property provides a bound for each of those objectives. We show that those bounds nicely combine to a global bound on the approximation ratio giving a constant-factor approximation with respect to a combinatorial bound and a fractional bound both under the original, general weight objective.

Theorem 13. *Let* $\mathcal{I}$ *be an instance of* Ordered k-Median *with a constant number of different weights in* w*. There exists a randomized algorithm for* Ordered k-Median *on* $\mathcal{I}$ *that computes expected* 38*-approximate solution in polynomial time.*

A straightforward application of this approach incorporating weight bucketing gives only a quasi-polynomial time algorithm (Theorem 15) due to the guessing part.

Theorem 15. *For any* $\epsilon > 0$*, there exists a randomized algorithm for* Ordered k-Median *that computes expected* $38(1+\epsilon)$*-approximate solution in quasi-polynomial time (specifically* $(nm)^{\mathcal{O}(\log_{1+\epsilon} n)}$*).*

To achieve a truly polynomial time algorithm we apply a clever distance bucketing approach by Aouad and Segev [7], which guesses for each distance bucket the average weight applied to this bucket by some optimal solution. Our analysis approach applies also to this more intricate setting but turns out technically more involved.

Relation to the Works of Chakrabarty and Swamy [41, 43], [42, 44].

Soon after the submission of our paper [35, 36], Chakrabarty and Swamy [41, 43] announced constant-factor approximation algorithms for Rectangular Ordered k-Median and also for Ordered k-Median. The result for Rectangular Ordered k-Median appears to be obtained independently. Instead of the LP-rounding process of Charikar and Li [48], they either use a primal-dual approach or a black-box reduction to k-Median.

A few months later Chakrabarty and Swamy [42, 44] improved an approximation constant for Ordered k-Median to $5+\epsilon$. One of the crucial differences is designing a deterministic LP-rounding procedure while we constructed a randomized one.

1.3.2 Harmonic k-Median and OWA k-Median

We introduce a general unified framework for two classes of problems: (i) extensions of the k-Median problem, where clients care about having multiple facilities in their vicinity, and (ii) finding winning committees according to a number of well-known, but hard-to-compute multiwinner election rules. Let us first formalize our framework defining the OWA k-Median problem.

In the OWA k-Median problem we have the same input as in k-Median and also our goal is to choose a k-subset of facilities. The difference is in the objective function: in k-Median the cost of a client depends only on the closest opened facility, but in Ordered k-Median the cost depends on all open facilities. Following Yager [149], we use ordered weighted average (OWA) operators to define the cost of a client for a bundle of k facilities $\mathcal{W}$. Formally, let $w^k = (w_1^k, \ldots, w_k^k)$ be a non-increasing vector of k weights. We define the w^k-cost of a client j for a k-size set of facilities $\mathcal{W}$ as $w^k(\mathcal{W}, j) = \sum_{i=1}^{k} w_i^k c_i^{\rightarrow}(\mathcal{W}, j)$, where $c^{\rightarrow}(\mathcal{W}, j)$ is a non-decreasing permutation of the costs of client j for the facilities from $\mathcal{W}$. Informally speaking, the highest weight is applied to the lowest cost, the second highest weight to the second lowest cost, etc. In this thesis we study the following computational problem.

OWA k-Median
Input:
$\mathcal{F}, \mathcal{C}, k, c$: the same as in k-Median, but a cost function $c\colon \mathcal{F} \times \mathcal{C} \to \mathbb{R}_{\geq 0}$ can be general (not necessarily metric),
$w^k = (w_1^k, \ldots, w_k^k)$: a non-increasing weight vector, $w_i^k \geq 0$.
Notation:
$c^{\rightarrow}(\mathcal{W}, j) = (c_1^{\rightarrow}(\mathcal{W}, j), \ldots, c_k^{\rightarrow}(\mathcal{W}, j))$ is a non-decreasing permutation of the costs from $\{c_{ij}\colon i \in \mathcal{W}\}$ for $j \in \mathcal{C}$ and $\mathcal{W} \subseteq \mathcal{F}, |\mathcal{W}| = k$.
Output: A set $\mathcal{W} \subseteq \mathcal{F}, |\mathcal{W}| = k$ that minimizes *the total connection cost*:

$$\sum_{j \in \mathcal{C}} \sum_{i=1}^{k} w_i^k c_i^{\rightarrow}(\mathcal{W}, j).$$

We stress the differences with the definition of Ordered k-Median. Here w^k is a vector of k elements (in contrast to w that has n elements). In OWA k-Median the

vector of weights w^k is applied to a sorted vector of k connection costs of a particular client j to all open facilities $\mathcal{W}$. In Ordered k-Median each client is connected only to the closest open facility, hence we consider n connection costs that are multiplied by a vector w of n elements. In OWA k-Median we consider general cost functions (not necessarily metric).

Note that OWA k-Median in a metric space with weights $(1, 0, \ldots, 0)$ is exactly the k-Median problem. We are specifically interested in the following two sequences of weights:

(1) **harmonic:** $(1, 1/2, 1/3, \ldots, 1/k)$. By the Harmonic k-Median problem we denote the OWA k-Median problem with the harmonic vector of weights.

(2) **p-geometric:** $(1, p, p^2, \ldots, p^{k-1})$, for some $p < 1$.

The two aforementioned sequences of weights, harmonic and p-geometric, have their natural interpretations, which we discuss later on (for instance, see Examples 1 and 2).

In the following two subsections we discuss the applicability of the studied model in two settings: multiwinner elections and clustering/facility location.

Multiwinner Elections

Different variants of the OWA k-Median problem are very closely related to the preference aggregation methods and multiwinner election rules studied in the computational social choice, in particular, and in AI, in general—we summarize this relation in Table 1.2 and in Figure 1.1. In particular, one can observe that each "median" problem is associated with a corresponding "winner" problem. Let us now explain the differences between the winner ("election") and the median ("centroid clustering") problems:

1. The election problems are usually formulated as maximization problems, where instead of (negative) costs we have (positive) utilities. The two variants, the minimization (with costs) and the maximization (with utilities) have the same optimal solutions. Yet, there is a substantial difference in their approximability.

   Approximating the minimization variant is usually much harder. For instance, consider the Chamberlin–Courant (CC) rule which is defined by using the sequence of weights $(1, 0, \ldots, 0)$. In the maximization variant standard arguments can be used to prove that a greedy procedure yields the approximation ratio of $(1 - 1/e)$. This stands in sharp contrast to the case when the same rule is expressed as the minimization one; in such a case we cannot hope for virtually any approximation [137]. Approximating the minimization variant is also more desired. E.g., a $1/2$-approximation algorithm for (maximization) CC can effectively ignore half of the population of voters, whereas it was argued [137] that a 2-approximation algorithm for the minimization (if existed) would be more powerful. In this thesis we study the harder minimization variant, and give the first constant-factor approximation algorithm for the minimization OWA-Winner with the harmonic weights.

centroid clustering problem	multiwinner election rule	comment
OWA k-Median (this thesis)	OWA-Winner [136]	Finding winners according to OWA-Winner rules is the maximization variant of OWA k-Median (utilities instead of costs).
	Thiele methods [147]	Thiele methods are OWA-Winner rules for binary costs.
Harmonic k-Median (this thesis)	Proportional Approval Voting (PAV) [147]	PAV is the maximization variant of Harmonic k-Median and in PAV we assume additionally binary costs.
k-Median [79, 80, 91]	Chamberlin–Courant (CC) [45]	In CC, usually some specific form of utilities is assumed—different utilities have been considered, but always in the maximization variant (utilities instead of costs).

Table 1.2: The relation between clustering problems and the corresponding multiwinner election problems studied in AI, in particular in the computational social choice community.

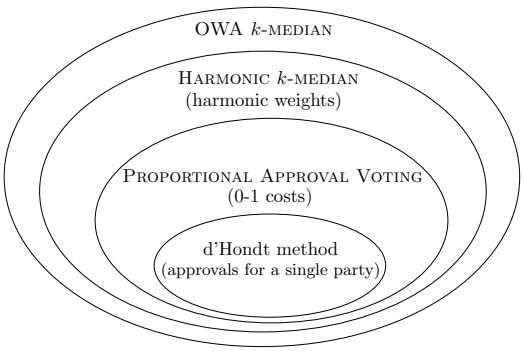


Figure 1.1: The relationship between the considered models. OWA k-Median is the most general model. Proportional Approval Voting and Harmonic k-Median due to the use of harmonic weights can be viewed as natural extensions of the well-known and commonly used D'Hondt method of apportionment [30].

2. In clustering problems it is usually assumed that the costs satisfy the triangle inequality. This relates to the previous point: since the problem cannot be well approximated in the general setting, one needs to make additional assumptions. One of our main results is showing that there is a k-median problem (Harmonic k-Median) that admits a constant-factor approximation without assuming that the costs satisfy the triangle inequality; to the best of our knowledge this is the first known result of this kind.

The special case of Harmonic k-Median where each cost belongs to the binary set $\{0, 1\}$ is equivalent to finding winners according to Proportional Approval Voting (PAV). The harmonic sequence $(1, 1/2, 1/3, \ldots, 1/k)$ is in a way exceptional: indeed, PAV can be viewed as an extension of the well-known D'Hondt method of apportionment (used for electing parliaments in many contemporary democracies) to the case where the voters can vote for individual candidates rather than for political parties [30]. Further, PAV satisfies several other appealing properties, such as extended justified representation [12]. This is one of the reasons why we are specifically interested in the harmonic weights. For more discussion on PAV and other

approval-based rules, we refer the reader to the survey of Kilgour [97].

OWA k-Median as an Extension of k-Median

Intuitively, our general formulation extends k-Median to scenarios where the clients not only use their most preferred facilities, but when there exists a more complex relationship of "using the facilities" by the clients. Similar intuition is captured by the Fault Tolerant version of the k-Median problem introduced by Swamy and Shmoys [143] and recently studied by Hajiaghayi et al. [78]. There, the idea is that the facilities can be malfunctioning, and to increase the resilience to their failures each client needs to be connected to several of them. The Fault Tolerant k-Median problem is defined as follows.

Fault Tolerant k-Median
Input: The same as in k-Median, and additionally
r_j: a positive integer as *the connectivity requirement* of client $j \in \mathcal{C}$, $r_j \leq k$.
Notation:
$r_j(\mathcal{W})$: the sum of r_j largest values from $\{c_{i,j} : i \in \mathcal{W}\}$ for $j \in \mathcal{C}$ and $\mathcal{W} \subseteq \mathcal{F}, |\mathcal{W}| = k$.
Output: A set $\mathcal{W} \subseteq \mathcal{F}, |\mathcal{W}| = k$ that minimizes *the total connection cost*:
$$\sum_{j \in \mathcal{C}} r_j(\mathcal{W}).$$

When the values $(r_j)_{j \in \mathcal{C}}$ are all the same, i.e., if $r_j = r$ for all j, then Fault Tolerant k-Median is called the r-Fault Tolerant k-Median problem and it can be expressed as OWA k-Median for the weight vector w^k with r ones followed by $k - r$ zeros. Yet, in the typical setting of k-Median problems one assumes that the costs between clients and facilities behave like distances, i.e., that they satisfy the triangle inequality. Indeed, the $(2.675+\epsilon)$-approximation algorithm for k-Median [31], the 93-approximation algorithm for Fault Tolerant k-Median [78], the 2-approximation algorithm for k-Center [82], and the $(9+\epsilon)$-approximation algorithm for k-Means [1], they all use the triangle inequality. Moreover it can be shown by straightforward reductions from the Set Cover problem that there are no constant factor approximation algorithms for all these settings with general (non-metric) connection costs unless $\mathsf{P} = \mathsf{NP}$.

Using harmonic or geometric OWA weights is also well-justified in case of facility location problems, as illustrated by the following examples.

Example 1 (Harmonic weights: proportionality)**.** *Assume there are $\ell \leq k$ cities, and for $i \in \{1, 2, \ldots, \ell\}$ let N_i denote the set of clients who live in the i-th city. For the sake of simplicity, let us assume that $k \cdot |N_i|$ is divisible by n. Further, assume that the cost of traveling between any two points within a single city is negligible (equal to zero), and that the cost of traveling between different cities is equal to one. Our goal is to decide in which cities the k facilities should be opened; naturally, we set the cost of a client for a facility opened in the same city to zero, and—in another city—to one. Let us consider the* Harmonic k-Median *problem. Let n_i denote the*

number of facilities opened in the i-th city in the optimal solution. We will show that for each i we have $n_i = \frac{k|N_i|}{n}$, i.e., that the number of facilities opened in each city is proportional to its population. Towards a contradiction assume there are two cities, i and j, with $n_i \geq \frac{k|N_i|}{n} + 1$ and $n_j \leq \frac{k|N_j|}{n} - 1$. By closing one facility in the i-th city and opening one in the j-th city, we decrease the total cost by at least:

$$|N_j| \cdot w^k_{n_j+1} - |N_i| \cdot w^k_{n_i} = \frac{|N_i|}{n_j + 1} - \frac{|N_i|}{n_i} > \frac{|N_j|n}{k|N_j|} - \frac{|N_i|n}{k|N_i|} = 0.$$

Since we decreased the cost of the clients, this could not be an optimal solution. As a result we see that indeed for each i we have $n_i = \frac{k|N_i|}{n}$.

Example 2 (Geometric weights: probabilities of failures)**.** *Assume that we want to select k facilities and that each client will be using his or her favorite facility only. Yet, when a client wants to use a facility, it can be malfunctioning with some probability p; in such a case the client goes to her second most preferred facility; if the second facility is not working properly, the client goes to the third one, etc. Thus, a client uses her most preferred facility with probability $1-p$, her second most preferred facility with probability $p(1-p)$, the third one with probability $p^2(1-p)$, etc. As a result, the expected cost of a client j for the bundle of k facilities $\mathcal{W}$ for the weight vector $(1-p, (1-p)p, \ldots, (1-p)p^{k-1})$ is equal to*

$$\sum_{j \in \mathcal{C}} \sum_{i=1}^{k} (1-p)p^{i-1} c^{\rightarrow}_i(\mathcal{W}, j).$$

Therefore finding a set of facilities that minimize the expected cost of all clients is equivalent to solving OWA k-MEDIAN *for the p-geometric sequence of weights (in fact, the sequence that we use is a p-geometric sequence multiplied by $(1-p)$, yet multiplication of the weight vector by a constant does not influence the structure of the optimal solutions).*

Our Results and Techniques

Our main result is a 2.36-approximation algorithm for the HARMONIC k-MEDIAN problem (Theorem 23). We do not assume the triangle inequality. This is in contrast to the innaproximability of most clustering settings with general connection costs.

Theorem 23. *There exists a randomized algorithm for* HARMONIC k-MEDIAN *that computes expected 2.36-approximate solution in polynomial time.*

Our algorithm is based on dependent rounding of a solution to a natural linear program (LP) relaxation of the problem. We use the *dependent rounding* (DR) studied by Srinivasan et al. [75, 139], which transforms in a randomized way a fractional vector into an integral one. The sum-preservation property of DR ensures that exactly k facilities are opened.

DR satisfies what is well known as *negative correlation (NC)*—intuitively, this implies that the sums of subsets of random variables describing the outcome are more

centered around their expected values than if the fractional variables were rounded independently. More precisely, negative correlation allows one to use standard concentration bounds such as the Chernoff-Hoeffding bounds. Yet, interestingly, we find out that NC is not sufficient for our analysis in which we need a conditional variant of the concentration bounds. The property that is sufficient for conditional bounds is *negative association* (NA) [87]. In fact its special case that we call *binary negative association* (BNA), is sufficient for our analysis. It captures the capability of reasoning about conditional probabilities. Thus, our work demonstrates how to apply the (B)NA property in the analysis of approximation algorithms based on DR. To the best of our knowledge, Harmonic k-Median is the first natural computational problem, where it is essential to use BNA in the analysis of the algorithm. In Section 3.1 we discuss in detail the process of dependent rounding (including a few illustrative examples).

We additionally show in Theorem 29 that the 93-approximation algorithm for Fault Tolerant k-Median of Hajiaghayi et al. [78] can be extended to Metric OWA k-Median (OWA k-Median where the costs satisfy the triangle inequality).

Theorem 29. *There exists a randomized algorithm for* Metric OWA k-Median *that computes expected* 93*-approximate solution in polynomial time.*

The additional assumption for the costs is crucial. Indeed, OWA k-Median is hard to approximate with any constant for a large class of weight vectors; for instance, for p-geometric sequences with $p < 1/e$ (Theorem 22 and Corollary 23 in [32]) or for sequences where there exists $\lambda \in (0, 1)$ such that clients care only about the λ-fraction of opened facilities (Theorem 21 in [32]).

1.3.3 Minimax Approval Voting

We restrict our attention to approval-based multiwinner rules, i.e., rules where each voter expresses his or her preferences by providing a subset of the candidates which he or she approves. Various voting rules are studied in the literature [13, 26]. In the simplest one, Approval Voting, occurrences of each candidate are counted and k most often approved candidates are selected. While this rule has many desirable properties in the single winner case [74], in the multiwinner scenario its merits are often considered less clear [102], e.g., because it fails to reflect the diversity of interests in the electorate [97]. Therefore, numerous alternative rules have been proposed, including Satisfaction Approval Voting, Proportional Approval Voting, and Reweighted Approval Voting (for details see book chapter [97]).

We study a multiwinner voting rule called Minimax Approval Voting (MAV), introduced by Brams et al. [27]. Here, we see the votes and the choice as binary strings of length m (characteristic vectors of the subsets, i.e., the candidate i is approved if the string contains 1 at position i). For two strings x and y of the same length the Hamming distance $\mathcal{H}(x, y)$ is the number of positions where x and y differ, e.g., $\mathcal{H}(011, 101) = 2$. In MAV, we look for a binary string with k ones that minimizes the maximum Hamming distance to a vote. In other words, MAV minimizes the disagreement with the least satisfied voter and thus it is highly egalitarian: no voter is

ignored and a majority of voters cannot guarantee a specific outcome [103, 142]. John Rawls in his classic book "A Theory of Justice" [131] states that welfare is maximized when the utility of those society members that have the least is the greatest, so MAV is a *Rawlsian social welfare function*. For more general studies on minisum (utilitarian) and minimax (egalitarian) objectives see the work of Brams et al. [142]. Other egalitarian voting rules have been also studied by Betzler et al. [20].

Much recent research has been devoted to the axiomatic properties of multiwinner voting rules [65, 71, 133]. The goal is to classify and describe properties of voting rules because different voting rules have different properties, thus they should be used in proper scenarios. Let us look at some properties of MAV. MAV is not a proportional type of voting rule. A large group of voters with similar preferences can be not represented by the number of committee members proportional to the size of the group. Formally, it was shown that MAV does not satisfy *Justified Representation* property [12]. MAV is not strategy-proof [39], i.e., voters can vote strategically to be less dissatisfied by the outcome. On the other hand MAV *supports strong monotonicity with population increase* and *weak monotonicity without population increase* [133], and it is insensitive to clones [97].

MAV could be used when, for example, proportionality is not needed, agents are not selfish, but the outcome is required to be acceptable by every voter. A natural such scenario is when a group of experts have to make a decision or a group of friends want to choose activities for a holiday. Brams et al. [142, page 403] "commend the minimax procedure to colleges, universities, and other organizations".

We focus on the computational complexity of computing the choice based on the MAV rule. We consider an optimization problem called Minimax Approval Voting which is defined as follows (in a natural way we also define a decision version).

Minimax Approval Voting

Input:

$S = \{s_1, \ldots, s_n\}$: a multiset of binary strings of length m (also called votes),
k: a positive integer as the number of winners to choose.

Notation:

$x[i]$: is an i-th letter in a string x,
$\mathcal{H}(x, y) := \{i : x[i] \neq y[i]\}$ is the Hamming distance for two strings x and y.

Output: A string $s \in \{0, 1\}^m$ with exactly k ones that minimizes:

$$\max_{i \in \{1,2,\ldots,n\}} \mathcal{H}(s, s_i).$$

A reader familiar with string problems might recognize that Minimax Approval Voting is closely related to the classical NP-hard problem called Closest String which is studied in bioinformatics as a string problem with motivation in DNA-sequence-related topics and in the context of coding theory [106].

CLOSEST STRING
Input:
$S = \{s_1, \ldots, s_n\}$: a multiset of m-length strings over an alphabet Σ.
Notation:
$x[i]$: is an i-th letter in a string x,
$\mathcal{H}(x, y) := \{i : x[i] \neq y[i]\}$ is the Hamming distance for two strings x and y.
Output: A string $s \in \Sigma^m$ that minimizes:

$$\max_{i \in \{1,2,\ldots,n\}} \mathcal{H}(s, s_i).$$

Indeed, LeGrand et al. [104] showed that MINIMAX APPROVAL VOTING is NP-hard as well by reduction from CLOSEST STRING with a binary alphabet. The first proof of NP-hardness of MINIMAX APPROVAL VOTING was shown using reduction from VERTEX COVER [103]. This motivated the study on MINIMAX APPROVAL VOTING in terms of approximability and fixed-parameter tractability.

Previous Results on MINIMAX APPROVAL VOTING

The first approximation result for MINIMAX APPROVAL VOTING was a simple 3-approximation algorithm due to LeGrand et al. [104], obtained by choosing an arbitrary vote and taking any k approved candidates from the vote (extending it arbitrarily to k candidates if needed). Next, a 2-approximation was shown by Caragiannis et al. [39] using a deterministic LP-rounding procedure. They solve the natural LP relaxation of the problem and round the k highest values to 1 and the rest to 0. Obtained approximation ratio matches the integrality gap of the LP. This is achieved, for example, when all strings of length m with exactly $m/2$ ones are given as an input and $k = m/2$. Then the optimal solution of the LP is a vector of halves, hence the LP solution is located exactly in the middle of pairs of complementary strings.

In the area of fixed parameter tractability (FPT) every instance I of a decision problem P contains additionally an integer r, called a *parameter*. The goal is to find a *fixed parameter algorithm* (also called FPT algorithm), i.e., an algorithm with running time of the form $f(r)(|I|)^{\mathcal{O}(1)}$, where f is a computable function, which is typically at least exponential for NP-complete problems. If such an algorithm exists, we say that the problem P parameterized by r is fixed parameter tractable (FPT). For more details about FPT algorithms see the textbook of Cygan et al. [54] or the survey of Bredereck et al. [29] (in the context of computational social choice). The study of FPT algorithms for MINIMAX APPROVAL VOTING was initiated by Misra et al. [116]. They show, for example, that MINIMAX APPROVAL VOTING parameterized by k (the number of ones in the solution) is $W[2]$-hard, which implies that it does not admit an FPT algorithm, unless there is a highly unexpected collapse in parameterized complexity classes. From a positive perspective, they show that the problem is FPT when parameterized by the number of votes n or by the maximum allowed

distance d. Their algorithm runs in time[4] $\mathcal{O}^\star(d^{2d})$. [5] For a study on FPT complexity of generalizations of Minimax Approval Voting see the work of Baumeister et al. [16].

Previous Results on Closest String

It is interesting to compare the known results on Minimax Approval Voting with the corresponding ones on the better researched Closest String. The first PTAS for Closest String was given by Li et al. [106] with the running time bounded by $n^{\mathcal{O}(1/\epsilon^4)}$ where n is the number of the input strings. This was later improved by Andoni et al. [6] to $n^{\mathcal{O}(\frac{\log 1/\epsilon}{\epsilon^2})}$, and then by Ma and Sun [113] to $n^{\mathcal{O}(1/\epsilon^2)}$.

The first FPT algorithm for Closest String, running in time $\mathcal{O}^\star(d^d)$ was given by Gramm et al. [77]. This was later improved by Ma and Sun [113], who gave an algorithm with running time $\mathcal{O}^\star(2^{\mathcal{O}(d)} \cdot |\Sigma|^d)$, which is more efficient for constant-size alphabets. Further substantial progress is unlikely, since Lokshtanov et al. [110] have shown that Closest String admits no algorithms running in time $\mathcal{O}^\star(2^{o(d \log d)})$ or $\mathcal{O}^\star(2^{o(d \log |\Sigma|)})$, unless the Exponential Time Hypothesis (ETH) [84] fails.

Open Questions

The discrepancy between the state of the art for Closest String and Minimax Approval Voting raises interesting questions.

First, does the additional constraint on the number of ones in Minimax Approval Voting really make the problem harder? Is Minimax Approval Voting hard to approximate below 2 as it is for k-Center [82]? Note that Minimax Approval Voting (as well as Closest String) is a special case of the 1-Center problem on a hypercube that has an exponentially large set of facilities (2^m) but the input size is only nm.

Similarly, although in Minimax Approval Voting the alphabet is binary, no $\mathcal{O}^\star(2^{\mathcal{O}(d)})$-time algorithm is known, in contrast to Closest String. Can we find such an algorithm? In this thesis we answer the question on approximability of Minimax Approval Voting by presenting the following results.

Our Results and Techniques

We show the first polynomial time approximation scheme (PTAS) for Minimax Approval Voting (Theorem 39) which improves the previous best 2-approximation [39].

Theorem 39. *For any $\epsilon > 0$ we can find $(1+\epsilon)$-approximation solution for* Minimax Approval Voting *in polynomial time $n^{\mathcal{O}(1/\epsilon^4)} \cdot m^{\mathcal{O}(1)} + n^{\mathcal{O}(1/\epsilon)} \cdot m^{\mathcal{O}(1/\epsilon^4)}$ with probability at least $1-p$, for any fixed $p > 0$.*

[4] The $\mathcal{O}^\star$ notation suppresses factors polynomial in the input size.

[5] Actually, Misra et al. [116] claim the slightly better running time of $\mathcal{O}^\star(d^d)$. However, there is a flaw in the analysis [108, 115]: it states that the initial solution v is at distance at most d from the solution, while it can be at distance $2d$ because of, what we call here, the k-completion operation. This increases the maximum depth of the recursion to d (instead of the claimed $d/2$).

Our approximation scheme is based on the PTAS for Closest String [106]. Technically, our contribution is the method of handling the number of ones in the output. We also believe that our presentation is somewhat more intuitive.

The general idea behind our PTAS is to find a small enough subset X of votes that is a "good representation" of the whole set of votes S. Then the candidates are partitioned into those for which voters in X agree and the remaining candidates. For the "consensus candidates" we fix our decision to the decision induced by votes in X (additionally correcting the number of selected candidates in the "consensus" set). Next, we consider the optimization problem of finding a proper subset of the remaining candidates to join the committee. The key insight is that there exists a small enough subset X such that the induced decision for the "consensus candidates" will not be a big mistake.

For a description of our further results, let us recall the Exponential Time Hypothesis (ETH) of Impagliazzo and Paturi [84]. ETH states that there exists a constant $c > 0$, such that there is no algorithm solving 3-CNF-SAT in time $\mathcal{O}^\star(2^{cn})$, where n is the number of variables in the given 3-CNF-SAT instance. In recent years, ETH became the central conjecture used for proving tight bounds on the complexity of various problems, see the survey of Lokshtanov et al. [109]. Nevertheless, ETH-based lower bounds seem largely unexplored in the area of computational social choice [122]. Cygan et al. [56] showed that, unless ETH fails, there is no algorithm for Minimax Approval Voting running in time $\mathcal{O}^\star(2^{o(d \log d)})$. The result uses a reduction from the $k \times k$-Clique problem. Socała [138] provides an alternative reduction directly from $(3,4)$-CNF-SAT. As a corollary, the algorithm of Misra et al. [116] is essentially optimal, and indeed, in this sense Minimax Approval Voting is harder than Closest String.

Motivated by this, we show a parameterized approximation scheme for the decision version of Minimax Approval Voting.

Theorem 40. *There exists a randomized algorithm which, given an instance* $(S = \{s_i\}_{i\in[n]}, k, d)$ *of the decision version of* Minimax Approval Voting *(*d *is the required maximal distance) and any* $\epsilon \in (0,3)$*, runs in time* $\mathcal{O}\left(\left(\frac{3}{\epsilon}\right)^{2d} \cdot (m+n) + mn\right)$ *and either*

(*i*) *reports a solution at a distance at most* $(1+\epsilon)d$ *from* S*, or*

(*ii*) *reports that there is no solution at a distance at most* d *from* S*.*

In the latter case, the answer is correct with probability at least $1-p$*, for arbitrarily small fixed* $p > 0$*.*

Our algorithm uses a branching tree technique similarly to the $\mathcal{O}^\star(d^{2d})$-time algorithm of Misra et al. [116] but instead of considering all possible branches for a swap of 0 and 1 (deterministically) we sample uniformly at random a pair for a swap. We show that while there can be only one "good" branch for an optimal solution there is a constant fraction (dependent on ϵ) of "good" branches for the $(1+\epsilon)$-approximate solution.

Note that the lower bound of Cygan et al. [56] implies that, under (a randomized version of) ETH, this is essentially optimal, i.e., there is no parameterized approximation scheme running in time $\mathcal{O}^\star(2^{o(d\log(1/\epsilon))})$. Indeed, if such an algorithm existed, by picking $\epsilon = 1/(d+1)$ we would get an exact algorithm which contradicts our lower bound.

Finally, we get a faster PTAS for Minimax Approval Voting.

Theorem 46. *For any $\epsilon > 0$ we can find $(1+\epsilon)$-approximation solution for* Minimax Approval Voting *in polynomial time $n^{\mathcal{O}(1/\epsilon^2 \cdot \log(1/\epsilon))} \cdot m^{\mathcal{O}(1)}$ with probability at least $1 - r$, for any fixed $r > 0$.*

The idea is to use our parameterized approximation scheme for small values of an optimal solution. For large enough values of an optimal solution we use independent rounding of a solution to the natural LP relaxation of the problem and apply the Chernoff-Hoeffding bounds which might be of independent interest (Theorem 43).

The new PTAS is much faster than the previous one. In particular, the new running time does not contain the $m^{\mathcal{O}(1/\epsilon^4)}$ term, so one should expect a considerable speed-up when the number of votes is large. The running time of our faster PTAS almost matches the one of the fastest known PTAS for Closest String (up to a $\log(1/\epsilon)$ factor in the exponent).

1.4 Research Papers Being a Base of the Thesis

This thesis is based on the results contained in the following papers:

- All results contained in Chapter 2 were published in a conference paper:
  Jarosław Byrka, Krzysztof Sornat and Joachim Spoerhase.
  Constant-Factor Approximation for Ordered k-Median. STOC 2018 [36].

- A preliminary version of the results contained in Chapter 3 appeared in a conference paper:
  Jarosław Byrka, Piotr Skowron, Krzysztof Sornat.
  Proportional Approval Voting, Harmonic k-Median, and Negative Association. ICALP 2018 [33].

- A preliminary version of the first PTAS for Minimax Approval Voting contained in Section 4.1 appeared in a conference paper:
  Jarosław Byrka, Krzysztof Sornat.
  PTAS for Minimax Approval Voting. WINE 2014 [34].

- A parameterized approximation scheme and a faster PTAS for Minimax Approval Voting, contained in Section 4.2 and 4.3 respectively, were published in a journal paper:
  Marek Cygan, Łukasz Kowalik, Arkadiusz Socała and Krzysztof Sornat.
  Approximation and Parameterized Complexity of Minimax Approval Voting. Journal of Artificial Intelligence Research 63, 2018 [56].
  A preliminary version of both results appeared in a conference paper:
  Marek Cygan, Łukasz Kowalik, Arkadiusz Socała and Krzysztof Sornat.
  Approximation and Parameterized Complexity of Minimax Approval Voting. AAAI 2017 [55].

During my PhD studies I also worked on other topics which resulted in the following papers:

- Georgios Amanatidis, Evangelos Markakis and Krzysztof Sornat.
  Inequity Aversion Pricing over Social Networks: Approximation Algorithms and Hardness Results. MFCS 2016 [5].
  The paper contains approximation algorithms and APX-hardness proof for a social network (graph) problem called Inequity Aversion Pricing.

- Piotr Faliszewski, Pasin Manurangsi and Krzysztof Sornat.
  Approximation and Hardness of Shift-Bribery. AAAI 2019 [69].
  The paper contains the first PTAS for the Shift-Bribery problem with general positional scoring rules and inapproximability results for the Copeland$^{\alpha}$-Shift-Bribery problem assuming ETH or Gap-ETH.

Chapter 2

Ordered k-Median

In this chapter we show the first constant-factor approximation algorithm for Ordered k-Median (Theorem 16) which improves the previous best $\mathcal{O}(\log n)$-approximation due to Aouad and Segev [7].

In Section 2.1 we describe the algorithmic framework used in this chapter.

In Section 2.2 we show a 15-approximation algorithm for a special case called Rectangular Ordered k-Median (Theorem 5) which generalizes k-Median and k-Center. Then, we show a surprising *modularity* of Charikar and Li's rounding process (Lemma 11). The solution computed by this process is related to *all* rectangular objectives.

In Section 2.3 we *decompose* an arbitrary non-increasing weight vector into a convex combination of rectangular objectives. The aforementioned modularity property provides a bound for each of those objectives. This gives a polynomial time algorithm for Ordered k-Median with a constant number of different weights (Theorem 13) Then, we use weight bucketing to get a quasi-polynomial time algorithm due to the guessing part (Theorem 15).

In Section 2.4 we apply a clever distance bucketing approach by Aouad and Segev [7] with technically more involved analysis. This results in a polynomial time algorithm for Ordered k-Median (Theorem 16).

Additional notation for this chapter.

$\mathrm{cost}(\mathcal{W}) := \sum_{j=1}^{n} w_j \cdot c_j^{\rightarrow}(\mathcal{W})$ is the total connection cost in Ordered k-Median objective for a solution $\mathcal{W} \subseteq \mathcal{F}$.

$\mathrm{cost}_\ell(\mathcal{W}) := \sum_{j=1}^{\ell} c_j^{\rightarrow}(\mathcal{W})$ is the total connection cost in Rectangular Ordered k-Median objective with parameter ℓ (i.e. $w_\ell = 1$ and $w_{\ell+1} = 0$) for a solution $\mathcal{W} \subseteq \mathcal{F}$.

In $\mathrm{cost}(\cdot)$ and $\mathrm{cost}_\ell(\cdot)$ notation we ommit $w, c, \mathcal{C}$ and $c, \mathcal{C}$ respectively because they can be deduced from the context.

Let $j \in \mathcal{C}$ be a client. Then $\mathcal{B}(j, r)$ denotes the set of all facilities $i \in \mathcal{F}$ with $c_{ij} < r$, that is, $\mathcal{B}(j, r)$ is an open ball (in the set of facilities) of radius r around j.

We will assume w.l.o.g. that $w_1 = 1$ in the definition of Ordered k-Median.

Definition 1. *Consider an instance of* Ordered k-Median. *A* reduced cost function c^{r} *is a (not necessarily metric) function* $c^{\mathrm{r}}\colon \mathcal{D}\times\mathcal{D}\to\mathbb{R}_{\geq 0}$ *such that for all* $i,i',j,j'\in\mathcal{D}$ *we have that* $c^{\mathrm{r}}_{ij}\leq c_{ij}$ *and that* $c_{ij}\leq c_{i'j'}$ *implies* $c^{\mathrm{r}}_{ij}\leq c^{\mathrm{r}}_{i'j'}$.

Reduced cost functions arise naturally for Ordered k-Median since in its objective function non-increasingly sorted distances are multiplied by non-increasing weights which are smaller or equal to 1.

2.1 Algorithmic Framework

In what follows we describe the algorithmic framework used in this chapter. Some parts of this framework will be tailored to specific settings and are thus described later.

Our algorithms consist of two parts: an *LP-solving* and an *LP-rounding* part.

In the *LP-solving* part, we compute an optimal solution to an LP-relaxation, which is (apart from the objective function) identical to the standard LP relaxation for k-Median. However, instead of using the input metrics c in the objective function, we employ a reduced cost function c^{r}. Intuitively, in c^{r} the distances are multiplied by roughly the same weights as in a guessed optimal solution.

In the *LP-rounding* part the fractional solution provided by the above-described guessing will be rounded to an integral solution by applying the algorithm of Charikar and Li [48]. In contrast to the LP-solving part, this algorithm operates, however, in the *original* metric space rather than in the (generally non-metric) reduced cost space.

2.1.1 LP-Relaxation

Let $\mathrm{LP}(c^{\mathrm{r}})$ be the following relaxation of a natural ILP formulation of k-Median under some reduced cost function c^{r}.

$$\text{minimize} \quad \sum_{i\in\mathcal{F},j\in\mathcal{C}} c^{\mathrm{r}}_{ij}x_{ij} \qquad \text{s.t.} \tag{2.1}$$

$$x_{ij}\leq y_i \qquad i\in\mathcal{F}, j\in\mathcal{C} \tag{2.2}$$

$$\sum_{i\in\mathcal{F}} x_{ij} = 1 \qquad j\in\mathcal{C} \tag{2.3}$$

$$\sum_{i\in\mathcal{F}} y_i = k \tag{2.4}$$

$$0\leq x_{ij}, y_i\leq 1 \qquad i\in\mathcal{F}, j\in\mathcal{C} \tag{2.5}$$

Here, y_i denotes how much facility i is open (0—closed, 1—opened) and x_{ij} indicates how much client j is served by facility i (0—non-served, 1—served). Equality (2.4) ensures that exactly k facilities are opened (possibly fractionally), (2.3) guarantee that each client is served (possibly fractionally). (2.2) do not allow a facility to serve a client more than how much it is opened. For each client $j\in\mathcal{C}$ let $c^{\mathrm{r}}_{\mathrm{av}}(j)=\sum_{i\in\mathcal{F}} c^{\mathrm{r}}_{ij}x_{ij}$ denote the fractional (or average) reduced connection cost of j.

2.1.2 Guessing and LP-Solving

Note that if $c^r = c$ where c is the input metrics, LP(c) becomes the standard LP relaxation for the classical k-MEDIAN objective. In order to obtain a valid lower bound LP(c^r) for an ORDERED k-MEDIAN instance, we employ guessing of certain distances in an optimal solution. The details of the guessing are setting-specific and are thus described later.

Below, we describe some basic normalization steps for a feasible solution (x, y) to LP(c^r).

Definition 2. *Let (x, y) be a feasible solution to LP(c^r) where c^r is some reduced cost function. We call the assignment x of clients to facilities* distance-optimal *if x minimizes $\sum_{i\in\mathcal{F}, j\in\mathcal{C}} c_{ij}x_{ij}$ when y is kept fixed.*

Lemma 3. *We can w.l.o.g. assume that an optimal solution (x, y) to LP(c^r) for some reduced cost function c^r satisfies the following properties.*

(i) For any facility $i \in \mathcal{F}$ we have $y_i > 0$,

(ii) for any $i \in \mathcal{F}, j \in \mathcal{C}$ we have $x_{ij} \in \{0, y_i\}$,

(iii) the assignment x is distance-optimal.

Proof. To see the third property fix the opening vector y and some client j. Now sort all facilities i in non-decreasing order of their distance c_{ij} to j and greedily assign as much of the remaining demand of j to the current facility i (respecting the constraint $x_{ij} \leq y_i$). Stop when the full demand of j is served and repeat this process for all clients. Since the reduced cost function c^r respects the order of the original distances (see definition) the resulting assignment is optimal also under the reduced cost function.

The first and second properties are folklore and can be achieved by removing or duplicating facilities (see [48]). □

We define $\mathcal{F}_j = \{i \in \mathcal{F} : x_{ij} > 0\}$. For $\mathcal{F}' \subseteq \mathcal{F}$ we define the *volume* of $\mathcal{F}'$ as $\text{vol}(\mathcal{F}') = \sum_{i\in\mathcal{F}'} y_i$. Note that $\text{vol}(\mathcal{F}_j) = 1$ for any feasible solution.

2.1.3 LP-Rounding: Dependent Rounding Approach of Charikar and Li

We round the fractional solution obtained in the LP-solving phase to an integral solution by the (slightly modified) LP-rounding process of Charikar and Li [48] for k-MEDIAN.

To apply this algorithm note that the feasibility of a solution (x, y) to LP(c^r) does *not* depend on the cost vector c^r. This enables us to compute an optimum solution (x, y) to LP(c^r) for some appropriate reduced cost function and to subsequently apply the rounding process of Charikar and Li (which operates on the original metrics c) to the solution (x, y). In the analysis, we have to exploit how c^r and c are related in order to bound the approximation ratio of the algorithm.

The rounding algorithm of Charikar and Li consists of four phases: a clustering phase, a bundling phase, a matching phase, and a sampling phase (see Algorithm 1).

Algorithm 1: Rounding Algorithm by Charikar and Li [48]

Data: feasible fractional solution (x, y) to LP(c^{r}) satisfying the properties of Lemma 3
Result: set of k facilities

```
/* Clustering phase */
/* run a clustering procedure to compute a set 𝒞′ ⊆ 𝒞 of cluster
centers so that each client j ∈ 𝒞 is "close" to some cluster center
j′ ∈ 𝒞′ and so that the cluster centers are "far" from each other */
```
$\mathcal{C}' \leftarrow \mathrm{Clustering}(x, y)$;
```
/* Bundling phase */
```
for $j \in \mathcal{C}'$ **do**
  $R_j \leftarrow \frac{1}{2} \min_{j' \in \mathcal{C}', j' \neq j}(c_{jj'})$;
  $\mathcal{U}_j \leftarrow \mathcal{F}_j \cap \mathcal{B}(j, R_j)$;
```
/* Matching phase */
```
$\mathcal{M} \leftarrow \emptyset$;
while *there are unmatched clients in* $\mathcal{C}'$ **do**
  add to $\mathcal{M}$ a pair from $\mathcal{C}'$ that is the closest pair among unmatched clients in $\mathcal{C}'$
```
/* Sampling phase (dependent rounding) */
/* Apply dependent randomized rounding as described by Charikar and Li
[48] preserving the marginals for the individual facilities, bundles,
matched pairs in ℳ, and set ℱ */
```
return $\mathrm{DependentRounding}(x, y, \{\mathcal{U}_j\}_j, \mathcal{M}, \mathcal{F})$

Below we give some intuition on the different phases. More formal arguments will be given later.

The purpose of the *clustering procedure* is to compute a set $\mathcal{C}' \subseteq \mathcal{C}$ of *cluster centers* so that each client $j \in \mathcal{C}$ is "close" to some cluster center $j' \in \mathcal{C}'$ and so that the cluster centers are "far" from each other. We thus may think of the cluster centers representing all remaining clients. The implementation of the procedure and the meaning of "close" and "far" is application-specific and will thus be described later.

In the *bundling phase* each cluster center $j \in \mathcal{C}'$ is associated with a bundle $\mathcal{U}_j$ of facilities. We will show that the volume of each bundle is at least 1/2 and that they are pairwise disjoint[1].

In the *matching phase* cluster centers are paired in a greedy manner. The total volume of the bundles of a matched pair is at least 1. This will ensure that in the

[1] We employ a simple unoptimized version of the argument, which is sufficient for constant-factor approximations. In the original optimized version the bundles are based on larger balls that may overlap but are made disjoint in a greedy manner. This allows Charikar and Li to obtain an improved approximation factor.

subsequent *sampling phase* at least one facility per pair is opened.

In the sampling phase we use the dependent randomized rounding procedure described by Charikar and Li [48] to open facilities and obtain a feasible solution. The procedure satisfies the following properties (as in the original work of Charikar and Li):

Lemma 4. *Let (x, y) be a feasible solution to LP(c^r) and assume that* $\mathrm{vol}(\mathcal{U}_j) \geq 1/2$ *and $\mathcal{U}_j \cap \mathcal{U}_{j'} = \emptyset$ for all distinct $j, j' \in \mathcal{C}'$. There is an efficient, randomized implementation of procedure* DependentRounding *in Algorithm 1 such that the following holds.*

(i) Each facility $i \in \mathcal{F}$ is opened with probability precisely y_i,

(ii) in each bundle $\mathcal{U}_j$ with $j \in \mathcal{C}'$ a facility is opened with probability precisely $\mathrm{vol}(\mathcal{U}_j)$*,*

(iii) for each matched pair (j, j') in $\mathcal{M}$ at least one facility in $\mathcal{U}_j \cup \mathcal{U}_{j'}$ will be opened,

(iv) in total at most k facilities are opened.

It is used that (x, y) is a (not necessarily optimal) feasible solution to LP(c), $\mathrm{vol}(\mathcal{U}_j \cup \mathcal{U}_{j'}) \geq 1$ for all distinct $j, j' \in \mathcal{C}'$, and that the union of set families $\{\{y_i\}\}_{i \in \mathcal{F}}$, $\{\mathcal{U}_j\}_{j \in \mathcal{C}'}$, $\{\mathcal{U}_j \cup \mathcal{U}_{j'} \mid (j, j') \in \mathcal{M}\}$, $\{\mathcal{F}\}$ forms a laminar family. The laminarity follows from the construction in the algorithm. The property $\mathrm{vol}(\mathcal{U}_j \cup \mathcal{U}_{j'}) \geq 1$ follows from the assumption $\mathrm{vol}(\mathcal{U}_j) \geq 1/2$ and $\mathcal{U}_j \cap \mathcal{U}_{j'} = \emptyset$ for all distinct $j, j' \in \mathcal{C}'$, which depends on the implementation of the clustering procedure and has thus to be proven for the specific implementation.

2.2 Rectangular Weight Vectors

In this section, we apply the algorithmic framework described in the previous section, to obtain a constant-factor approximation algorithm for Rectangular Ordered k-Median, which generalizes k-Median and k-Center. More specifically, we will show the following.

Theorem 5. *There exists a randomized algorithm for* Rectangular Ordered k-Median *that computes expected* 15*-approximate solution in polynomial time.*

To proof this theorem, we need to fill in the following two missing parts of the framework: guessing of the reduced cost space and the clustering procedure in the rounding part.

2.2.1 Guessing and Reduced Costs

In the LP-solving phase, we guess the value T of ℓ-th largest distance in an optimum solution to Rectangular Ordered k-Median. (This is the smallest distance that is counted in the total connection cost with non-zero weight.) As the correct guess

of T is the distance between a client and a facility the guessing can be performed by considering only $\mathcal{O}(mn)$ options for T.

For each $i \in \mathcal{F}, j \in \mathcal{C}$, we define the *reduced cost*

$$c_{ij}^T = \begin{cases} c_{ij} & \text{if} \quad c_{ij} \geq T, \\ 0 & \text{otherwise,} \end{cases} \tag{2.6}$$

that will be used as a cost function in our LP for the ORDERED k-MEDIAN.

An optimal solution (x, y) to LP(c^T) is a feasible solution for LP(c) as well. As introduced in Section 2.1.1, we use $c_{\text{av}}(j) = \sum_{i \in \mathcal{F}} x_{ij} \cdot c_{ij}$ and $c_{\text{av}}^T(j) = \sum_{i \in \mathcal{F}} x_{ij} \cdot c_{ij}^T$ to denote the average connection cost and the average reduced connection cost of a client $j \in \mathcal{C}$, respectively.

2.2.2 Dedicated Clustering

The following two clustering methods (Algorithms 2 and 3) will be considered. The algorithms differ only slightly and we have underlined the differences. We first analyze using Algorithm 2.

Algorithm 2: Dedicated Clustering

Data: feasible fractional solution (x, y) to LP(c)
Result: set $\mathcal{C}' \subseteq \mathcal{C}$ of cluster centers

$\mathcal{C}' \leftarrow \emptyset$;
$\mathcal{C}'' \leftarrow \mathcal{C}$;
$\underline{c_{\text{av}}^T(j)} \leftarrow \sum_{i \in \mathcal{F}} x_{ij} \cdot \underline{c_{ij}^T}$ for all $j \in \mathcal{C}$;
while $\mathcal{C}''$ *is non empty* **do**
    take $j \in \mathcal{C}''$ with the smallest $\underline{c_{\text{av}}^T(j)}$;
    add j to $\mathcal{C}'$;
    delete from $\mathcal{C}''$ client j;
    delete from $\mathcal{C}''$ all clients j' with $c_{jj'} \leq \underline{4c_{\text{av}}^T(j') + 4T}$
return $\mathcal{C}'$

Algorithm 3: Oblivious Clustering

Data: feasible fractional solution (x, y) to LP(c)
Result: set $\mathcal{C}' \subseteq \mathcal{C}$ of cluster centers

$\mathcal{C}' \leftarrow \emptyset$;
$\mathcal{C}'' \leftarrow \mathcal{C}$;
$\underline{c_{\text{av}}(j)} \leftarrow \sum_{i \in \mathcal{F}} x_{ij} \cdot \underline{c_{ij}}$ for all $j \in \mathcal{C}$;
while $\mathcal{C}''$ *is non empty* **do**
    take $j \in \mathcal{C}''$ with the smallest $\underline{c_{\text{av}}(j)}$;
    add j to $\mathcal{C}'$;
    delete from $\mathcal{C}''$ client j;
    delete from $\mathcal{C}''$ all clients j' with $c_{jj'} \leq \underline{4c_{\text{av}}(j')}$
return $\mathcal{C}'$

This clustering procedure is very similar to the one of Charikar and Li (see also Section 2.2.4 below) except for the fact that the procedure needs to know the threshold T of the guessing phase[2]. This dependence allows a simpler and better analysis for Rectangular Ordered k-Median. In Section 2.2.4, we will describe how to get rid of this dependency, which allows us to generalize the result.

2.2.3 Analysis of the Algorithm

In the following we analyze Algorithm 1 using the procedure Dedicated Clustering.

Observation 6. *We have $c^T_{ij} \leq c_{ij} \leq c^T_{ij} + T$ for any $i \in \mathcal{F}, j \in \mathcal{C}$ and consequently $c^T_{\text{av}}(j) \leq c_{\text{av}}(j) \leq c^T_{\text{av}}(j) + T$.*

The following two lemmas and their proofs are modifications of the corresponding claims by Charikar and Li [48].

Lemma 7. *The following two statements are true for Algorithm 1 with Dedicated Clustering .*

(i) For any $j, j' \in \mathcal{C}'$ we have that $c_{jj'} > 4\max(c^T_{\text{av}}(j), c^T_{\text{av}}(j')) + 4T$.

(ii) For any $j \in \mathcal{C} \setminus \mathcal{C}'$ there is a client $j' \in \mathcal{C}'$ with $c^T_{\text{av}}(j') \leq c^T_{\text{av}}(j)$ and $c_{jj'} \leq 4c^T_{\text{av}}(j) + 4T$.

Proof. To see (i) assume w.l.o.g. that j is considered before j' as a potential cluster center in the algorithm. Thus $c^T_{\text{av}}(j) \leq c^T_{\text{av}}(j')$. If $c_{jj'} \leq 4c^T_{\text{av}}(j') + 4T = 4\max(c^T_{\text{av}}(j), c^T_{\text{av}}(j')) + 4T$ then j' would be deleted from $\mathcal{C}''$ when j is considered. A contradiction to the fact that j' is a cluster center.

In order to see (ii), consider an arbitrary client $j \in \mathcal{C} \setminus \mathcal{C}'$. As j is not a cluster center it was deleted from $\mathcal{C}''$ when some cluster center $j' \in \mathcal{C}'$ was considered. For this cluster center we have $c_{jj'} \leq 4c^T_{\text{av}}(j) + 4T$. □

Lemma 8. *The following two statements are true for Algorithm 1 with Dedicated Clustering.*

(i) $\text{vol}(\mathcal{U}_j) \geq 0.5$ for all $j \in \mathcal{C}'$.

(ii) $\mathcal{U}_j \cap \mathcal{U}_{j'} = \emptyset$ for all $j, j' \in \mathcal{C}'$, $j \neq j'$.

Proof. To prove statement (i) consider an arbitrary $j \in \mathcal{C}'$. Let $j' \in \mathcal{C}'$ be such that $2R_j = c_{jj'}$. We have $c_{jj'} > 4c^T_{\text{av}}(j) + 4T \geq 4c_{\text{av}}(j)$ and hence, $R_j > 2c_{\text{av}}(j)$. Therefore, $c_{\text{av}}(j) = \sum_{i \in \mathcal{F}_j} x_{ij} c_{ij} \geq \sum_{i \in \mathcal{F}_j \setminus \mathcal{U}_j} x_{ij} c_{ij} \geq R_j \cdot \sum_{i \in \mathcal{F}_j \setminus \mathcal{U}_j} x_{ij} \geq R_j \cdot \text{vol}(\mathcal{F}_j \setminus \mathcal{U}_j)$ where the last inequality follows because $x_{ij} = y_i$ for all $i \in \mathcal{F}$ and $j \in \mathcal{C}'$. Therefore $\text{vol}(\mathcal{F}_j \setminus \mathcal{U}_j) < 1/2$ and $\text{vol}(\mathcal{U}_j) > 1/2$.

To prove (ii) consider distinct $j, j' \in \mathcal{C}'$. By the definition of R_j we have $c_{jj'} \geq 2R_j$. Hence, for any facility i in $\mathcal{B}(j, R_j)$ we have $c_{ij} < c_{ij'}$, which implies (ii). □

[2]Note that we use T explicitly but also implicitly in the average reduced cost $c^T_{\text{av}}(j)$.

We are now ready to prove the main result of this section.

Proof of Theorem 5. Let $\mathcal{W}_{\text{OPT}}$ be an optimum integer solution under the objective cost_ℓ, let (x, y) be the optimum (fractional) solution to $\text{LP}(c^T)$, and let A be the (random) solution output by Algorithm 1. Let $\text{OPT} = \text{cost}_\ell(\mathcal{W}_{\text{OPT}})$, $\text{OPT}^* = \sum_{i\in\mathcal{F},j\in\mathcal{C}} c^T_{ij}x_{ij}$, and $\text{ALG} = \text{cost}_\ell(A)$ be the values of an optimal solution, $\text{LP}(c^T)$ and Algorithm 1, respectively. Note that $\text{OPT}^* \leq \text{OPT}$ because $\mathcal{W}_{\text{OPT}}$ can be interpreted as a feasible solution to $\text{LP}(c^T)$.

Let $j \in \mathcal{C}$ be a client and let C_j be the random variable denoting the distance (according to the original metrics c) traveled by j in A. The idea of the analysis is to define two separate budgets (random variables) D_j and X_j that together give an upper bound on C_j, that is, $C_j \leq D_j + X_j$. The budget D_j is "deterministically" set to $5T$ and does not depend on the random choices of the algorithm. The "probabilistic" budget X_j is a random variable (depending on the random choices made by the algorithm) that is constructed in an incremental way below. We will show below that (by suitably constructing X_j) the connection cost C_j of j can actually be upper bounded by $D_j + X_j$ and that $\mathbb{E}[X_j] \leq 10 \cdot c^T_{\text{av}}(j)$. We claim that this will complete our proof of a 15-approximation. To this end, note that at most ℓ clients j contribute their deterministic budget D_j to $\text{cost}_\ell(\cdot)$ because at most ℓ distances are actually accounted for in the objective function. Unfortunately, an analogous reasoning does not hold true for the expected value of the random variables X_j. (For example, note that $\mathbb{E}[\max(X_1, \ldots, X_n)]$ is generally unbounded in $\max(\mathbb{E}[X_1], \ldots, \mathbb{E}[X_n])$ in the case of $\ell = 1$.) However, we can just sum over *all* those random variables obtaining the following upper bound on the total expected connection cost:

$$\mathbb{E}[\text{ALG}] \leq D_j \cdot \ell + \sum_{j\in\mathcal{C}} \mathbb{E}[X_j] \leq 5\ell \cdot T + 10 \cdot \sum_{j\in\mathcal{C}} c^T_{\text{av}}(j) \leq 15 \cdot \text{OPT}.$$

For the last inequality, note that by our guess of T we have that $\text{OPT} \geq \ell \cdot T$ and from the definition of $\text{LP}(c^T)$ we have $\text{OPT}^* = \sum_{j\in\mathcal{C}} c^T_{\text{av}}(j)$. To establish that $C_j \leq D_j + X_j$ consider an arbitrary client j with connection cost C_j. We incrementally construct our upper bound on C_j starting with 0. Each increment will be either charged to D_j or X_j.

Consider a client j and the cluster center j' it is assigned to (possibly $j = j'$). We have that $c_{jj'} \leq 4c^T_{\text{av}}(j) + 4T$ by Lemma 7 (ii). We charge $4T$ to D_j and $4c^T_{\text{av}}(j)$ with probability 1 to X_j.

We now describe how to pay for the transport from j' to an open facility. There are two cases to distinguish. Either a facility within a radius T around j' is opened or not. If yes, then this cost can be covered by charging an additional amount of T to D_j. In this case the total cost is upper bounded by $D_j = 5T$ plus X_j where we have $\mathbb{E}[X_j] = 4c^T_{\text{av}}(j) \leq 10c^T_{\text{av}}(j)$ as desired.

If no facility within a radius T around j' is opened then observe that for each facility i with $c_{ij'} \geq T$ we have that $c^T_{ij'} = c_{ij'}$. We now continue to bound the connection cost for this case. Let j'' be the closest client distinct from j' in $\mathcal{C}'$. We consider the case where j' and j'' are not matched. (The case where they are matched is simpler.) Let j''' be the client in $\mathcal{C}'$ to which j'' is matched, i.e., $(j'', j''') \in \mathcal{M}$. By

the dependent rounding process one facility in $\mathcal{U}_{j''} \cup \mathcal{U}_{j'''}$ will be opened. We have that $c_{j'j''} = 2R_{j'} =: 2R$ (where R_j is defined as in Algorithm 1) and thus $c_{j''j'''} \leq 2R$ and $R_{j''}, R_{j'''} \leq R$ (otherwise, j'' would not have been matched with j''' but with j').

This means that, in case no facility is opened in the bundle $\mathcal{U}_{j'}$ the client j travels an additional distance of at most $\max(c_{j'j''}+R_{j''}, c_{j'j''}+c_{j''j'''}+R_{j'''}) \leq 2R+2R+R \leq 5R$.

If a facility is opened in the bundle $\mathcal{U}_{j'}$ then we charge this additional connection cost to X_j. The contribution of this case to $\mathbb{E}[X_j]$ is (by Properties (i) and (ii) of Lemma 4) at most

$$\sum_{i \in \mathcal{U}_{j'} \setminus \mathcal{B}(j',T)} y_i c_{ij'} = \sum_{i \in \mathcal{U}_{j'} \setminus \mathcal{B}(j',T)} x_{ij'} c_{ij'} = \sum_{i \in \mathcal{U}_{j'} \setminus \mathcal{B}(j',T)} x_{ij'} c^T_{ij'}$$
$$\leq \sum_{i \in \mathcal{F}_{j'}} x_{ij'} c^T_{ij'} = c^T_{\mathrm{av}}(j')\,.$$

Here, the first equality follows by our assumption $x_{ij} \in \{0, y_i\}$ from Section 2.1.2. The second equality follows because we assume that no facility is opened in $\mathcal{B}(j',T)$ and since $c_{ij'} = c^T_{ij'}$ for all $i \in \mathcal{U}_{j'} \setminus \mathcal{B}(j',T)$.

We finally handle the case where no facility in $\mathcal{U}_{j'}$ is opened and where j additionally travels a distance of at most $5R$. We charge this additional cost to X_j. We bound the probability that this case occurs. We claim that $\mathrm{vol}(\mathcal{U}_{j'})$ is at least $1 - c^T_{\mathrm{av}}(j')/R$. To see this, recall that $2R \geq c_{j''j'''} > 4\max(c^T_{\mathrm{av}}(j''), c^T_{\mathrm{av}}(j''')) + 4T$ thus $R > T$. Note that the reason of adding the quantity $4T$ in the clustering phase (Algorithm 2, line 8) is to have the property $R > T$ (in the original algorithm of Charikar-Li [48] this property is not necessarily satisfied). Using this, for all facilities in $\mathcal{F}_{j'} \setminus \mathcal{U}_{j'}$ we have that $c^T_{ij'} = c_{ij}$ because $R > T$. Hence

$$c^T_{\mathrm{av}}(j') \geq \sum_{i \in \mathcal{F}_{j'} \setminus \mathcal{B}(j',T)} x_{ij'} \cdot c^T_{ij'} = \sum_{i \in \mathcal{F}_{j'} \setminus \mathcal{B}(j',T)} x_{ij'} \cdot c_{ij'} \geq \sum_{i \in \mathcal{F}_{j'} \setminus \mathcal{U}_{j'}} x_{ij'} \cdot c_{ij'}$$
$$\geq R \cdot \sum_{i \in \mathcal{F}_{j'} \setminus \mathcal{U}_{j'}} x_{ij'} = R \cdot \mathrm{vol}(\mathcal{F}_{j'} \setminus \mathcal{U}_{j'}) = R \cdot \Big(1 - \mathrm{vol}(\mathcal{U}_{j'})\Big),$$

which implies the claim. Here, note that $\mathcal{B}(j',T) \subseteq \mathcal{U}_{j'}$ because $R > T$. This means that j travels the additional distance of $5R$ with probability at most $c^T_{\mathrm{av}}(j')/R$ and hence the contribution to $\mathbb{E}[X_j]$ is upper bounded by $5 \cdot c^T_{\mathrm{av}}(j')$. Summarizing, for the case when no facility is opened within $\mathcal{B}(j',T)$ we can upper bound $\mathbb{E}[X_j]$ by:

- a cost of serving client j through the closest cluster center j' that is $4 \cdot c^T_{\mathrm{av}}(j)$, plus
- a value $c^T_{\mathrm{av}}(j')$ for the case when a facility is opened within bundle $\mathcal{U}_{j'}$, plus
- a value $5R$ with probability at most $c^T_{\mathrm{av}}(j')/R$ when no facility is opened within $\mathcal{U}_{j'}$.

Hence $\mathbb{E}[X_j] \leq 4 \cdot c^T_{\mathrm{av}}(j) + 1 \cdot c^T_{\mathrm{av}}(j') + 5R \cdot c^T_{\mathrm{av}}(j')/R \leq 10 \cdot c^T_{\mathrm{av}}(j)$, by taking into account that $c^T_{\mathrm{av}}(j') \leq c^T_{\mathrm{av}}(j)$. Moreover we charged again at most $5T$ to D_j in this case. In the end we have the desired two upper bounds for both budgets for completing the proof: $D_j \leq 5T$, $\mathbb{E}[X_j] \leq 10 \cdot c^T_{\mathrm{av}}(j)$. □

2.2.4 Oblivious Clustering

In Algorithm 1, we are working on the original metrics c but still Dedicated Clustering described in the previous section depends on our guessed parameter T and the reduced metrics c^T. In this section, we show that we can apply the original clustering of Charikar and Li that works solely on the input metrics c and that is thus *oblivious* of the guessing phase. In particular, we use the Oblivious Clustering procedure as described in Algorithm 3. The following two lemmas that are analogous to Lemmas 7 and 8 are due to Charikar and Li [48]

Lemma 9. *The following two statements are true for Algorithm 1 with Oblivious Clustering.*

(i) *For any $j, j' \in \mathcal{C}'$ we have that* $c_{jj'} > 4\max(c_{\text{av}}(j), c_{\text{av}}(j'))$.

(ii) *For any $j \in \mathcal{C} \setminus \mathcal{C}'$ there is a client $j' \in \mathcal{C}'$ with* $c_{\text{av}}(j') \leq c_{\text{av}}(j)$ *and* $c_{jj'} \leq 4c_{\text{av}}(j)$.

Lemma 10. *The following two statements are true for Algorithm 1 with Oblivious Clustering.*

(i) $\text{vol}(\mathcal{U}_j) \geq 0.5$ *for all $j \in \mathcal{C}'$.*

(ii) $\mathcal{U}_j \cap \mathcal{U}_{j'} = \emptyset$ *for all $j, j' \in \mathcal{C}'$, $j \neq j'$.*

Using Oblivious Clustering, we can prove the following version of Theorem 5. While the constants proven in the following lemma are weaker than the ones for Dedicated Clustering, it exhibits a surprising modularity that is a key ingredient to later handle the general case. In particular, the clustering (and thus the whole rounding phase) are unaware (oblivious) of the cost vector $\bar{c}$ with respect to which we optimized $\text{LP}(\bar{c})$. Secondly, the bound proven in the lemma holds for *any* rectangular objective function of Ordered k-Median (specified by parameter ℓ), threshold T and the corresponding average reduced cost and may be unrelated to the cost function $\bar{c}$ that we optimized to obtain the fractional solution (x, y).

Lemma 11. *Consider a feasible fractional solution (x, y) to $LP(c)$ where x is distance-optimal. Let $\ell \geq 1$ be a positive integer, let $T \geq 0$ be arbitrary. Then we have* $\mathbb{E}[\text{cost}_\ell(A)] \leq 19\ell T + 19\sum_{j \in C} c^T_{\text{av}}(j)$ *where A is the (random) solution output by the Algorithm 1 with Oblivious Clustering.*

Proof. As in the proof of Theorem 5, we provide for each client an upper bound on the distance C_j (according to the original distance c) traveled by this client.

Again, the upper bound is paid for by two budgets D_j and X_j. The "deterministic" budget D_j is $19T$. The "probabilistic" budget X_j is a random variable (depending on the random choices made by the algorithm).

We will show below that (by a suitable choice of X_j) the connection cost C_j of j can actually be upper bounded by $D_j + X_j$ and $\mathbb{E}[X_j] \leq 19c^T_{\text{av}}(j)$. If this can be shown

then this will complete our proof of a constant-factor approximation. As before, note that at most ℓ clients j will pay the budget $D_j = 19T$ because at most ℓ distances are actually accounted for in the objective function. Analogously to the case of Dedicated Clustering, we obtain:

$$\mathbb{E}[\mathrm{ALG}] \leq D_j \cdot \ell + \sum_{j \in C} \mathbb{E}[X_j] \leq 19\ell \cdot T + 19 \cdot \sum_{j \in C} c^T_{\mathrm{av}}(j).$$

To show the claim consider an arbitrary client j with connection cost C_j. We incrementally construct our upper bound on C_j starting with 0. Each increment will be either charged to D_j or X_j.

Consider a client j and the cluster center j' it is assigned to (possibly $j = j'$). We have that $c_{jj'} \leq 4c_{\mathrm{av}}(j) \leq 4c^T_{\mathrm{av}}(j) + 4T$ by line 8 of Algorithm 3. We charge $4T$ to D_j and $4c^T_{\mathrm{av}}(j)$ with probability 1 to X_j.

We now describe how to pay for the transport from j' to an open facility. There are two cases to distinguish. Either a facility within a radius of βT is opened or not. (Here, $\beta \geq 2$ is a parameter to be determined later.) If yes, then this cost can be covered by charging an additional amount of βT to D_j. In this case the total cost is upper bounded by $D_j = (\beta + 4)T$ and $\mathbb{E}[X_j] = 4c^T_{\mathrm{av}}(j)$.

If *no* facility within a radius of βT of j' is opened then observe that for all facilities i with $c_{ij'} \geq \beta T$ we have that $c^T_{ij'} = c_{ij'}$ because of $\beta \geq 1$. We now continue to bound the connection cost for this case. Let j'' be the closest client distinct from j' in $\mathcal{C}'$. We consider the case where j'' and j' are not matched. (The case where they are matched is simpler.) Let j''' be the client in $\mathcal{C}'$ to which j'' is matched i.e. $(j'', j''') \in \mathcal{M}$. By the dependent rounding process one facility in $\mathcal{U}_{j''} \cup \mathcal{U}_{j'''}$ will be opened. We have that $c_{j'j''} = 2R_{j'} = 2R$ and thus $c_{j''j'''} \leq 2R$ and $R_{j''}, R_{j'''} \leq R$ (otherwise, j'' and j''' would not have been matched). This means that in case no facility is opened in the bundle $\mathcal{U}_{j'}$ the client j travels an additional distance (in expectation) of at most $\max(c_{j'j''} + R_{j''}, c_{j'j''} + c_{j''j'''} + R_{j'''}) \leq 2R + 2R + R \leq 5R$.

If a facility is opened in the bundle $\mathcal{U}_{j'}$ then we charge this additional connection cost to X_j. The contribution of the additional connection cost in this case to the expectation of X_j cost is at most

$$\sum_{i \in \mathcal{U}_{j'} \setminus \mathcal{B}(j', \beta T)} x_{ij'} c_{ij'} = \sum_{i \in \mathcal{U}_{j'} \setminus \mathcal{B}(j', \beta T)} x_{ij'} c^T_{ij'} \leq \sum_{i \in \mathcal{F}_{j'} \setminus \mathcal{B}(j', \beta T)} x_{ij'} c^T_{ij'}\,. \tag{2.7}$$

Here, equality holds because we assume that no facility is opened in $\mathcal{B}(j', \beta T)$ where $\beta \geq 1$ and because therefore $c_{ij'} = c^T_{ij'}$ for all $i \in \mathcal{U}_{j'} \setminus B(j', \beta T)$. The right hand side of (2.7) is denoted by $c^T_{\mathrm{far}}(j')$ and is clearly upper bounded by $\sum_{i \in \mathcal{F}_{j'}} x_{ij} c^T_{ij} = c^T_{\mathrm{av}}(j')$.

We finally handle the case where no facility in $\mathcal{U}_{j'}$ is opened and where j additionally travels a distance of at most $5R$. If $R \leq \beta T$, we can charge the additional travel distance of at most $5\beta T$ to D_j. Hence, we focus on the difficult case where $R > \beta T$ and where the maximum distance traveled can be unbounded in terms of T. We charge this additional cost to X_j. We bound the probability that this case occurs. We claim that $\mathrm{vol}(\mathcal{U}_{j'})$ is at least $1 - c^T_{\mathrm{far}}(j')/R$. To see this, note that for all

facilities in $\mathcal{F}_{j'} \setminus \mathcal{U}_{j'}$ we have that $c^T_{ij'} = c_{ij}$ because $R > \beta T$ and $\beta \geq 1$. Hence

$$c^T_{\text{far}}(j') = \sum_{i \in \mathcal{F}_{j'} \setminus \mathcal{B}(j', \beta T)} x_{ij'} \cdot c^T_{ij'} = \sum_{i \in \mathcal{F}_{j'} \setminus \mathcal{B}(j', \beta T)} x_{ij'} \cdot c_{ij'} \geq \sum_{i \in \mathcal{F}_{j'} \setminus \mathcal{U}_{j'}} x_{ij'} \cdot c_{ij'}$$
$$\geq R \cdot \sum_{i \in \mathcal{F}_{j'} \setminus \mathcal{U}_{j'}} x_{ij'} = R \cdot \text{vol}(\mathcal{F}_{j'} \setminus \mathcal{U}_{j'}) = R \cdot (1 - \text{vol}(\mathcal{U}_{j'})),$$

which implies the claim. Here, note that $\mathcal{B}(j', \beta T) \subseteq \mathcal{U}_{j'}$ because $R > \beta T$. This means that j travels the additional distance of at most $5R$ with probability at most $c^T_{\text{far}}(j')/R$ and hence the increment to X_j in expectation is upper bounded by $5 \cdot c^T_{\text{far}}(j')$. Thus, for the case where no facility is opened within a radius of βT around j', we can upper bound $\mathbb{E}[X_j]$ by:

- an expected cost of serving client j through the closest cluster center j' that is $4 \cdot c^T_{\text{av}}(j)$ (random part), plus
- a value $c^T_{\text{far}}(j')$ (with probability at most one), plus
- a value $5 \cdot R$ with probability at most $c^T_{\text{far}}(j')/R$.

Hence $\mathbb{E}[X_j] \leq 4 \cdot c^T_{\text{av}}(j) \cdot 1 + c^T_{\text{far}}(j') \cdot 1 + 5 \cdot R \cdot c^T_{\text{far}}(j')/R = 4c^T_{\text{av}}(j) + 5c^T_{\text{far}}(j')$. As in Oblivious Clustering we sort the clients according to c_{av} rather than c^T_{av} we do not necessarily have that $c^T_{\text{av}}(j')$ or even $c^T_{\text{far}}(j')$ are upper bounded by $c^T_{\text{av}}(j)$. We still can relate the latter two quantities in the following way.

First, assume that $c_{jj'} > \alpha T$ where $1 \leq \alpha < \beta - 1$ is a parameter to be determined later. We have that $c_{\text{av}}(j') \leq c_{\text{av}}(j)$ by our (oblivious) clustering. Hence $c^T_{\text{av}}(j') \leq c_{\text{av}}(j') \leq c_{\text{av}}(j) \leq c^T_{\text{av}}(j) + T$. On the other hand, $\alpha T < c_{jj'} \leq 4c_{\text{av}}(j)$ since j was assigned to j'. Hence $T < 4/\alpha \cdot c_{\text{av}}(j)$ and thus $c^T_{\text{av}}(j') \leq (1 + 4/\alpha)c^T_{\text{av}}(j)$. Since $c^T_{\text{far}}(j') \leq c^T_{\text{av}}(j')$ we can upper bound $\mathbb{E}[X_j]$ in this case by $(9 + 20/\alpha)c^T_{\text{av}}(j)$.

Second, assume that $c_{jj'} \leq \alpha T$. Recall that we assume further that no facility is opened within $\mathcal{B}(j', \beta T)$. We claim that in the assignment vector x the total demand assigned from j' to $\mathcal{F} \setminus \mathcal{B}(j', \beta T)$ is *at most* the total demand assigned from j to $\mathcal{F} \setminus \mathcal{B}(j', \beta T)$. This is, because any facility within the ball $\mathcal{B}(j', \beta T)$ is (trivially) strictly closer than any facility not in this ball. Hence, if j would manage to assign strictly more demand to facilities inside the ball than j' does, then we could construct a new assignment for j' that also serves strictly more demand of j' within this ball contradicting the optimality of x. Now, we are going to construct a (potentially suboptimal) assignment of the part of the demand of j' contributing to $c^T_{\text{far}}(j')$ that can be upper bounded in terms of $c^T_{\text{av}}(j)$. As the optimum assignment will clearly will have the same upper bound this will conclude our proof. To this end, we simply assign the demand of j' outside of the ball $\mathcal{B}(j', \beta T)$ in the same way as does j. Note that by our above claim this provides enough demand as j ships at least as much demand outside the ball as j' does. In particular let i be an arbitrary facility outside the ball. We now set $x'_{ij'} := x_{ij}$ to obtain our new assignment for j' Note that by the triangle inequality $c_{ij} \geq c_{ij'} - c_{jj'} \geq (\beta - \alpha)T \geq T$ and thus $c_{ij} = c^T_{ij}$ (a constraint

$\alpha \leq \beta - 1$ was introduced to obtain $c_{ij} = c^T_{ij}$ in this case). Therefore

$$\frac{c^T_{ij'}}{c^T_{ij}} = \frac{c_{ij'}}{c_{ij}} \leq \frac{c_{ij'}}{c_{ij'} - c_{jj'}} \leq \frac{\beta T}{(\beta - \alpha)T} = \frac{\beta}{\beta - \alpha}\,.$$

x' can be not optimal assignement for j', hence

$$\begin{aligned} c^T_{\text{far}}(j') &\leq \sum_{i \in \mathcal{F} \setminus \mathcal{B}(j', \beta T)} x'_{ij'} c^T_{ij'} \\ &\leq \frac{\beta}{\beta - \alpha} \sum_{i \in \mathcal{F} \setminus \mathcal{B}(j', \beta T)} x_{ij} c^T_{ij} \leq \frac{\beta}{\beta - \alpha} c^T_{\text{av}}(j)\,. \end{aligned}$$

In the end we have two upper bounds for both budgets:

$$\begin{aligned} D_j &\leq (4 + 5\beta)T, \\ \mathbb{E}[X_j] &\leq \max\left\{4 + \frac{5\beta}{\beta - \alpha}, 9 + \frac{20}{\alpha}\right\} c^T_{\text{av}}(j). \end{aligned}$$

Plugging $\alpha = 2$ and $\beta = 3$ gives the desired constants in the claim. □

By Lemma 11 we obtain that our algorithm with Oblivious Clustering yields a 38-approximation.

2.3 Handling the General Case

Consider an arbitrary instance of Ordered k-Median. Let w be the weight vector and let $\bar{w}$ the sorted weight vector using the same weights as w but without repetition. Let R be the number of distinct weight in both weight vectors. W.l.o.g. we assume that all distances c_{ij} for $i \in \mathcal{F}, j \in \mathcal{C}$ are pairwise distinct. (This can be achieved by slightly perturbing the input distances.) To apply our algorithmic framework, we guess *tresholds* T_r for $r = 1, \ldots, R$ such that T_r is the smallest distance c_{ij} that is multiplied by weight of value $\bar{w}_r$ in some fixed optimum solution. To guess the thresholds T_r we check $(nm)^R$ many candidates. Additionally, we define $T_0 = \infty$. We have $T_r < T_{r-1}$ for $r = 1, \ldots, R$ because we assumed pairwise distinct distances. For each $i \in \mathcal{F}, j \in \mathcal{C}$ we assign the connection cost c_{ij} to the weight $w(i, j) = \bar{w}_r$, where $T_r \leq c_{ij} < T_{r-1}$. This leads us to the following definition of our reduced cost function $c^{\text{r}}_{ij} = c_{ij} \cdot w(i, j)$ for all $i \in \mathcal{F}, j \in \mathcal{C}$. We compute an optimal solution (x, y) to $LP(c^{\text{r}})$ and apply Algorithm 1 to (x, y).

Lemma 12. *The above-described randomized algorithm for* Ordered k-Median *computes expected* 38*-approximate solution. The algorithm makes* $\mathcal{O}((nm)^R)$ *many calls to Algorithm 1 with Oblivious Clustering, where* R *is the number of distinct weights in the weight vector* w*.*

Proof. Let $A \subseteq \mathcal{F}$ be the (random) solution output by the algorithm. Let OPT be the cost of optimum solution. For each $r = 1, \dots, R$ let ℓ_r be the largest index such that $w_{\ell_r} = \bar{w}_r$. From Lemma 11 we have for all $r = 1, \dots, R$

$$\mathbb{E}[\mathrm{cost}_{\ell_r}(A)] \le 19 \cdot \ell_r T_r + 19 \cdot \sum_{j \in \mathcal{C}} c_{\mathrm{av}}^{T_r}(j). \tag{2.8}$$

We decompose $\mathrm{cost}(A)$ into rectangular "pieces" (additionally defining $\bar{w}_{R+1} = 0$)

$$\begin{aligned}
\mathbb{E}[\mathrm{cost}(A)] &= \mathbb{E}\left[\sum_{\ell=1}^{n} w_\ell \cdot \overrightarrow{c_\ell}(A)\right] = \mathbb{E}\left[\sum_{r=1}^{R}\sum_{s=1}^{\ell_r} (\bar{w}_r - \bar{w}_{r+1}) \cdot \overrightarrow{c_s}(A)\right] \\
&= \mathbb{E}\left[\sum_{r=1}^{R} (\bar{w}_r - \bar{w}_{r+1}) \cdot \mathrm{cost}_{\ell_r}(A)\right] = \sum_{r=1}^{R} (\bar{w}_r - \bar{w}_{r+1}) \cdot \mathbb{E}[\mathrm{cost}_{\ell_r}(A)] \\
&\overset{(2.8)}{\le} 19 \cdot \sum_{r=1}^{R} (\bar{w}_r - \bar{w}_{r+1}) \cdot \ell_r T_r + 19 \cdot \sum_{r=1}^{R}\sum_{j \in \mathcal{C}} (\bar{w}_r - \bar{w}_{r+1}) \cdot c_{\mathrm{av}}^{T_r}(j).
\end{aligned} \tag{2.9}$$

We will bound this in terms of OPT. We know that an optimal solution pays at least cost T_r for weight in w equal to $\bar{w}_r$ for $r = 1, \dots, R$. Therefore, defining $\ell_0 = 0$ we have

$$\begin{aligned}
\mathrm{OPT} &\ge \sum_{r=1}^{R} \bar{w}_r \cdot (\ell_r - \ell_{r-1}) T_r \\
&= \sum_{r=1}^{R} \bar{w}_r \cdot \ell_r T_r - \sum_{r=2}^{R} \bar{w}_r \cdot \ell_{r-1} T_r \\
&\ge \sum_{r=1}^{R} \bar{w}_r \cdot \ell_r T_r - \sum_{r=2}^{R} \bar{w}_r \cdot \ell_{r-1} T_{r-1} \\
&= \sum_{r=1}^{R} \bar{w}_r \cdot \ell_r T_r - \sum_{r=1}^{R} \bar{w}_{r+1} \cdot \ell_r T_r \\
&= \sum_{r=1}^{R} (\bar{w}_r - \bar{w}_{r+1}) \cdot \ell_r T_r.
\end{aligned} \tag{2.10}$$

Moreover, we have

$$\begin{aligned}
&\sum_{r=1}^{R}\sum_{j \in \mathcal{C}} (\bar{w}_r - \bar{w}_{r+1}) \cdot c_{\mathrm{av}}^{T_r}(j) \\
&= \sum_{r=1}^{R}\sum_{j \in \mathcal{C}}\sum_{i \in \mathcal{F}} (\bar{w}_r - \bar{w}_{r+1}) \cdot x_{ij} \cdot c_{ij}^{T_r} \\
&= \sum_{j \in \mathcal{C}}\sum_{i \in \mathcal{F}} x_{ij} \cdot \sum_{r=1}^{R} (\bar{w}_r - \bar{w}_{r+1}) \cdot c_{ij}^{T_r} \\
&= \sum_{j \in \mathcal{C}}\sum_{i \in \mathcal{F}} x_{ij} \cdot \sum_{r\colon \bar{w}_r \le w(i,j)}^{R} (\bar{w}_r - \bar{w}_{r+1}) \cdot c_{ij} \\
&= \sum_{j \in \mathcal{C}}\sum_{i \in \mathcal{F}} x_{ij} \cdot w(i,j) \cdot c_{ij} = \sum_{j \in \mathcal{C}}\sum_{i \in \mathcal{F}} x_{ij} \cdot c_{ij}^{\mathrm{r}} \overset{(2.1)}{\le} \mathrm{OPT}.
\end{aligned} \tag{2.11}$$

Thus, we finally have $\mathbb{E}[\text{cost}(A)] \overset{(2.9),(2.10),(2.11)}{\leq} 38 \cdot \text{OPT}$. □

Theorem 13. *Let $\mathcal{I}$ be an instance of* Ordered k-Median *with a constant number of different weights in w. There exists a randomized algorithm for* Ordered k-Median *on $\mathcal{I}$ that computes expected 38-approximate solution in polynomial time.*

Proof. We have $|\{w_j : j \in \{1, 2, \ldots, n\}\}| = \mathcal{O}(1)$. Therefore using Lemma 12 we get a solution after $(nm)^{\mathcal{O}(1)}$ many calls to Algorithm 1 with Oblivious Clustering. Hence, in total, it takes polynomial time. □

Using standard bucketing arguments and neglecting sufficiently small weights, we can "round" an arbitrary weight vector into a weight vector with only a logarithmic number of different weights losing a factor of $1 + \epsilon$ in approximation. Plugging this into Lemma 11, we can obtain a $38(1 + \epsilon)$-approximation algorithm for the general case in time $(nm)^{\mathcal{O}(\log_{1+\epsilon} n)}$. This is a standard bucketing approach but for the paper being self-containing we provide the following formal calculations.

In Lemma 14 we show how to reduce the number of different weights to at most $\mathcal{O}(\log_{1+\epsilon} n)$. Main idea of such reduction is partitioning an interval $[0, w_1]$ into buckets with geometrical step $1+\epsilon$. Solving such instance we lose factor $1+\epsilon$ on approximation because for α-approximation solution W^*_α for $\mathcal{I}^*$, optimal solution W^*_{OPT} for $\mathcal{I}^*$ and optimal solution W_{OPT} for $\mathcal{I}$ we have

$$\begin{aligned}&\text{cost}_{\mathcal{I}}(W^*_\alpha) \leq (1+\epsilon)\text{cost}_{\mathcal{I}^*}(W^*_\alpha) \leq (1+\epsilon)\alpha \cdot \text{cost}_{\mathcal{I}^*}(W^*_{\text{OPT}})\\ &\leq (1+\epsilon)\alpha \cdot \text{cost}_{\mathcal{I}^*}(W_{\text{OPT}}) \leq (1+\epsilon)\alpha \cdot \text{cost}_{\mathcal{I}}(W_{\text{OPT}}).\end{aligned}$$

Lemma 14. *Let $\mathcal{I} = (\mathcal{F}, \mathcal{C}, c, k, w)$ be an instance of* Ordered k-Median *and $\epsilon > 0$. There exists an instance $\mathcal{I}^* = (\mathcal{F}, \mathcal{C}, c, k, w^*)$ of* Ordered k-Median *such that for any solution $\mathcal{W} \subseteq \mathcal{F}, |\mathcal{W}| = k$ we have*

$$\text{cost}_{\mathcal{I}^*}(\mathcal{W}) \leq \text{cost}_{\mathcal{I}}(\mathcal{W}) \leq (1+\epsilon) \cdot \text{cost}_{\mathcal{I}^*}(\mathcal{W})$$

and w^* has at most $\mathcal{O}(\log_{1+\epsilon} n)$ different values, i.e., $|\{a : \exists j \quad w^*_j = a\}| \in \mathcal{O}(\log_{1+\epsilon} n)$.

Proof. We define w^* by

$$w^*_j = \begin{cases} w_1 & \text{for} \quad j = 1, \\ (1+\epsilon)^{\lfloor \log_{1+\epsilon} w_j \rfloor} & \text{for} \quad w_j > \frac{\epsilon w_1}{n} \text{ and } j \neq 1, \\ 0 & \text{for} \quad w_j \leq \frac{\epsilon w_1}{n}. \end{cases} \tag{2.12}$$

First inequality follows directly from the definition of w^*_j. For the second inequality

we have

$$\begin{aligned}
&\mathrm{cost}_{\mathcal{J}}(\mathcal{W}) = \sum_{j=1}^{n} w_j \cdot c_j^{\rightarrow}(\mathcal{W}) \\
&= w_1 \cdot c_1^{\rightarrow}(\mathcal{W}) + \sum_{\substack{j:\, w_j > \frac{\epsilon w_1}{n} \\ j \neq 1}} w_j \cdot c_j^{\rightarrow}(\mathcal{W}) + \sum_{\substack{j:\, w_j \leq \frac{\epsilon w_1}{n} \\ j \neq 1}} w_j \cdot c_j^{\rightarrow}(\mathcal{W}) \\
&\leq w_1^* \cdot c_1^{\rightarrow}(\mathcal{W}) + \sum_{\substack{j:\, w_j > \frac{\epsilon w_1}{n} \\ j \neq 1}} (1+\epsilon) \cdot w_j^* \cdot c_j^{\rightarrow}(\mathcal{W}) + \sum_{\substack{j:\, w_j \leq \frac{\epsilon w_1}{n} \\ j \neq 1}} \frac{\epsilon w_1}{n} \cdot c_j^{\rightarrow}(\mathcal{W}) \\
&\leq w_1^* \cdot c_1^{\rightarrow}(\mathcal{W}) + \sum_{\substack{j:\, w_j > \frac{\epsilon w_1}{n} \\ j \neq 1}} (1+\epsilon) \cdot w_j^* \cdot c_j^{\rightarrow}(\mathcal{W}) + \epsilon \cdot w_1^* \cdot c_1^{\rightarrow}(\mathcal{W}) \\
&= (1+\epsilon) \cdot \sum_{j:\, w_j > \frac{w_1}{n}} w_j^* \cdot c_j^{\rightarrow}(\mathcal{W}) = (1+\epsilon) \cdot \mathrm{cost}_{\mathcal{J}^*}(\mathcal{W}).
\end{aligned}$$

Let us assume that there is at least $2\log_{1+\epsilon}(n)+5$ different values w_j^* and n is large enough. We know that the highest value of w_j^* is equal to w_1. It is possible that the lowest value of w_j^* is equal to 0. By the induction we can show that the p-th highest value of $\{w_j^* : w_j^* > 0\}$ for $p \in \{2, 3, \ldots, \lceil 2\log_{1+\epsilon} n \rceil + 3\}$ is at most $\frac{w_1}{(1+\epsilon)^{p-2}}$. Therefore

$$\min\{w_j^* : w_j^* > 0\} < \frac{w_1}{(1+\epsilon)^{\lceil 2\log_{1+\epsilon} n \rceil + 3 - 2}} \leq \frac{w_1}{(1+\epsilon)n^2} \leq \frac{w_1}{(1+\epsilon)n}.$$

So there exists j such that $\frac{w_1}{n} \geq (1+\epsilon)w_j^* > 0$ and $w_j > \frac{\epsilon w_1}{n}$. From the definition we have $(1+\epsilon)w_j^* = (1+\epsilon)^{\lfloor \log_{1+\epsilon} w_j \rfloor + 1} > (1+\epsilon)^{\log_{1+\epsilon} w_j} = w_j > \frac{\epsilon w_1}{n} > \frac{w_1}{n}$. Contradiction. Therefore w^* has at most $\mathcal{O}(\log_{1+\epsilon} n)$ different values. □

Theorem 15. *For any $\epsilon > 0$, there exists a randomized algorithm for* Ordered k-Median *that computes expected $38(1+\epsilon)$-approximate solution in quasi-polynomial time (specifically $(nm)^{\mathcal{O}(\log_{1+\epsilon} n)}$).*

Proof. We transform a vector of weights w into w^* using Lemma 14. On that we lose $(1+\epsilon)$ to the approximation factor but we get an instance with only $\mathcal{O}(\log_{1+\epsilon} n)$ different weights. Then we apply Lemma 12. □

2.4 Polynomial-Time $(38+\epsilon)$-Approximation Algorithm

To obtain a truly-polynomial time algorithm we use the clever bucketing approach proposed by Aouad and Segev [7]. In this approach the *distances* are grouped into logarithmically many distance classes thereby losing a factor $1+\epsilon$. For each distance class the *average* weight is guessed up to a factor of $1+\epsilon$. The crucial point is that this guessing can be achieved in polynomial time because the average weights are non-decreasing with increasing distance class. This leads to a reduced cost function

based on average weights. The resulting analysis decomposes the weight vector into $n = |\mathcal{C}|$ many rectangular objectives. While the proof strategy is similar in spirit to the one of Lemma 12 it turns out to be technically more involved. In the remainder of this section we prove the following theorem.

Theorem 16. *For any $\epsilon > 0$, there exists a randomized algorithm for* Ordered k-Median *that computes expected $(38+\epsilon)$-approximate solution in polynomial time (specifically $(nm)^{\mathcal{O}(1/\epsilon \log(1/\epsilon))}$).*

2.4.1 Distance Bucketing

Let $\mathcal{W}_{\text{OPT}}$ be an optimal solution to a given Ordered k-Median instance. Let $c_{\max} := c_1^{\rightarrow}(\mathcal{W}_{\text{OPT}})$ be the maximum connection cost in this solution. We assume that we know $c_{\max}$ as it is one of $\mathcal{O}(mn)$ many possible distances in the input. Fix an error parameter $\epsilon > 0$ and let $c_{\min} := \epsilon \cdot c_{\max}/n$. Roughly speaking, distances smaller than $c_{\min}$ can have only negligible impact on any feasible solution as they may increase its cost by a factor of at most $1+\epsilon$.

We now partition the distances of the vector $c^{\rightarrow}(\mathcal{W}_{\text{OPT}})$ into $S := \lceil \log_{1+\epsilon}(n/\epsilon) \rceil = \mathcal{O}(\frac{1}{\epsilon} \log \frac{n}{\epsilon})$ many distance classes. More precisely, for all $s = 0, \ldots, S-1$ introduce the intervals $D_s = (c_{\max}(1+\epsilon)^{-(s+1)}, c_{\max}(1+\epsilon)^{-s}]$. Let $D_S = [0, c_{\max}(1+\epsilon)^{-S}] \ni c_{\min}$. For all $s = 0, \ldots, S$ let $J_s = \{\, j \mid c_j^{\rightarrow}(\mathcal{W}_{\text{OPT}}) \in D_s \,\}$ and let $C_s = \{\, c_j^{\rightarrow}(\mathcal{W}_{\text{OPT}}) \mid j \in J_S \,\}$. The classes $C_0, \ldots, C_S$ form a disjoint partition of $c^{\rightarrow}(\mathcal{W}_{\text{OPT}})$ where some of the classes may, however, be empty. For technical reasons, we assume that none of the intput distances $c_{ij}, i \in \mathcal{F}, j \in \mathcal{C}$ coincides with a boundary of one of the intervals D_s for some $s = 0, \ldots, S$. This can be achieved by slightly increasing all boundaries of the intervals using the fact that the intervals are left-open. Additionally we define $J_{\geq s} = \bigcup_{r=s}^{S} J_r$.

2.4.2 Guessing Average Weights

For any non-empty class C_s let

$$w_{\text{av}}^s := \frac{1}{|C_s|} \sum_{j \in J_s} w_j \tag{2.13}$$

denote the *average* weight applied to distances in this class. If C_s is empty then w_{av}^s denotes the smallest weight w_j applied to some distance $c_j^{\rightarrow}(\mathcal{W}_{\text{OPT}})$ in a non-empty class C_l with $l < s$. Such a class always exists as $C_0 \ni c_{\max}$ is non-empty.

As argued by Aouad and Segev [7], it is possible to guess the values of w_{av}^s up to a factor of $1+\epsilon$ in polynomial time $n^{\mathcal{O}(1/\epsilon \log 1/\epsilon)}$. This is, because we have $w_{\text{av}}^0 \geq w_{\text{av}}^1 \geq \cdots \geq w_{\text{av}}^S$ and because it suffices to guess those values as powers of $1+\epsilon$. More precisely, as a result of this we assume that we are given values $w_{\text{gs}}^0 \geq w_{\text{gs}}^1 \geq \cdots \geq w_{\text{gs}}^S$ with $w_{\text{av}}^s \leq w_{\text{gs}}^s \leq (1+\epsilon) w_{\text{av}}^s$ for $i = 0, \ldots, S$.

2.4.3 Reduced Cost Function and LP-Solving

We are now ready to define our reduced cost function. For all values of $d \in [0, c_{\max}]$ let $w(d)$ be the weight w^s_{gs} such that $d \in D_s$ for some $s \in \{0, \ldots, S\}$. For each $i \in \mathcal{F}, j \in \mathcal{C}$ such that $c_{ij} \leq c_{\max}$ let $c^{\text{r}}_{ij} := w(c_{ij}) \cdot c_{ij}$. Now solve the linear program $\text{LP}(c^{\text{r}})$ with additional constraints $x_{ij} = 0$ for all $i \in \mathcal{F}, j \in \mathcal{C}$ such that $c_{ij} > c_{\max}$. In what follows let (x, y) denote an optimal solution to this LP. Now apply the rounding algorithm of Charikar and Li with Oblivious Clustering (Algorithm 1 with clustering as in Algorithm 3) to obtain an integral solution $A \subseteq \mathcal{F}, |A| = k$.

Let OPT be the value (cost) of an optimum solution W_{OPT} for Ordered k-Median and let OPT^* be the value of an optimum solution for $LP(c^{\text{r}})$, let A be the solution for Ordered k-Median computed by our algorithm. We define distance class (interval) in which the distance d falls by $D(d)$ and $w_{n+1} = 0$.

Using Lemma 11 with $T_\ell = \max(D(c^{\rightarrow}_\ell(W_{\text{OPT}})))$ for each $\ell = 1, \ldots, n$ we obtain

$$\mathbb{E}[\text{cost}_\ell(A)] \leq 19 \cdot \ell T_\ell + 19 \cdot \sum_{j \in \mathcal{C}} c^{T_\ell}_{\text{av}}(j)\,. \tag{2.14}$$

We can partition the cost of our algorithm $\text{cost}(A)$ into rectangular pieces as follows

$$\begin{aligned}
\mathbb{E}[\text{cost}(A)] &= \mathbb{E}\left[\sum_{\ell=1}^{n} w_\ell \cdot c^{\rightarrow}_\ell(A)\right] = \mathbb{E}\left[\sum_{\ell=1}^{n}\sum_{r=1}^{\ell}(w_\ell - w_{\ell+1}) \cdot c^{\rightarrow}_r(A)\right] \\
&= \mathbb{E}\left[\sum_{\ell=1}^{n}(w_\ell - w_{\ell+1}) \cdot \text{cost}_\ell(A)\right] = \sum_{\ell=1}^{n}(w_\ell - w_{\ell+1}) \cdot \mathbb{E}\left[\text{cost}_\ell(A)\right] \\
&\overset{(2.14)}{\leq} 19 \cdot \sum_{\ell=1}^{n}(w_\ell - w_{\ell+1}) \cdot \ell T_\ell + 19 \cdot \sum_{\ell=1}^{n}\sum_{j \in \mathcal{C}}(w_\ell - w_{\ell+1}) \cdot c^{T_\ell}_{\text{av}}(j)\,.
\end{aligned} \tag{2.15}$$

We would like to upper bound this in terms of OPT. We know that the optimal solution pays at least cost $\inf(D_s)$ for each distance in distance bucket C_s and thus

$$\text{OPT} \geq \sum_{s=0}^{S}\left(\inf(D_s) \cdot \sum_{\ell \in J_s} w_\ell\right) = \sum_{s=0}^{S}\inf(D_s) \cdot w^s_{\text{av}} \cdot |C_s|. \tag{2.16}$$

Lemma 17.

$$\sum_{s=0}^{S}\inf(D_s) \cdot w^s_{\text{av}} \cdot |C_s| \geq \frac{1}{1+\epsilon}\sum_{\ell=1}^{n}(w_\ell - w_{\ell+1}) \cdot \ell \cdot T_\ell. \tag{2.17}$$

Proof. The right hand side is equal to

$$
\begin{aligned}
&\frac{1}{1+\epsilon}\sum_{\ell=1}^{n}(w_\ell - w_{\ell+1})\cdot\ell\cdot\max\Big(D(c_\ell^{\rightarrow}(W_{\mathrm{OPT}}))\Big)\\
\leq&\sum_{\ell=1}^{n}(w_\ell - w_{\ell+1})\cdot\ell\cdot\inf\Big(D(c_\ell^{\rightarrow}(W_{\mathrm{OPT}}))\Big)\\
=&\sum_{s=0}^{S}\sum_{\ell\in J_s}(w_\ell - w_{\ell+1})\cdot\ell\cdot\inf(D_s)\\
=&\sum_{\substack{s=0\\J_s\neq\emptyset}}^{S}\inf(D_s)\Bigg(w_{\min(J_s)}\cdot\min(J_s)+\sum_{\ell\in J_s\setminus\min(J_s)}w_\ell\cdot\ell\\
&\qquad\qquad-\sum_{\ell\in J_s\setminus\max(J_s)}w_{\ell+1}\cdot\ell - w_{\max(J_s)+1}\cdot\max(J_s)\Bigg)\\
=&\sum_{\substack{s=0\\J_s\neq\emptyset}}^{S}\inf(D_s)\Bigg(w_{\min(J_s)}\cdot\min(J_s)+\sum_{\ell\in J_s\setminus\min(J_s)}w_\ell\cdot\ell\\
&\qquad\qquad-\sum_{\ell\in J_s\setminus\min(J_s)}w_\ell\cdot(\ell-1) - w_{\max(J_s)+1}\cdot\max(J_s)\Bigg)\\
=&\sum_{\substack{s=0\\J_s\neq\emptyset}}^{S}\inf(D_s)\Bigg(w_{\min(J_s)}\cdot\Big(\min(J_s)-1\Big)+\sum_{\ell\in J_s}w_\ell - w_{\max(J_s)+1}\cdot\max(J_s)\Bigg)\\
=&\sum_{s=0}^{S}\inf(D_s)\cdot w_{\mathrm{av}}^{s}\cdot|C_s|\\
&+\sum_{\substack{s=0\\J_s\neq\emptyset}}^{S}\inf(D_s)\cdot\Big(w_{\min(J_s)}\cdot\Big(\min(J_s)-1\Big) - w_{\max(J_s)+1}\cdot\max(J_s)\Big).
\end{aligned}
$$

The proof ends with showing that the second factor is non-positive. We define $\mathcal{E}$ as a set of non-consecutive non-empty intervals such that all intervals between them are empty. Formally

$$
\begin{aligned}
\mathcal{E} = \Big\{&(s_1,s_2)\in\{1,\ldots,S\}^2:\\
&s_1+2\leq s_2 \quad\text{and}\quad J_{s_1},J_{s_2}\neq\emptyset \quad\text{and}\quad \forall_{s_3\in\{s_1,s_1+1,\ldots,s_2\}}J_{s_3}=\emptyset\Big\}.
\end{aligned}
$$

Then

$$
\begin{aligned}
&\sum_{\substack{s=0\\J_s\neq\emptyset}}^{S} \inf(D_s)\Big(w_{\min(J_s)}\cdot\Big(\min(J_s)-1\Big)-w_{\max(J_s)+1}\cdot\max(J_s)\Big)\\
&=\sum_{\substack{s=1\\J_s\neq\emptyset}}^{S} \inf(D_s)\cdot w_{\min(J_s)}\cdot\Big(\min(J_s)-1\Big)\\
&\quad-\sum_{\substack{s=0\\J_s\neq\emptyset}}^{S-1} \inf(D_s)\cdot w_{\max(J_s)+1}\cdot\max(J_s)\\
&=\sum_{\substack{s=1\\J_s\neq\emptyset}}^{S} \inf(D_s)\cdot w_{\min(J_s)}\cdot\Big(\min(J_s)-1\Big)\\
&\quad-\sum_{\substack{s=1\\J_{s-1}\neq\emptyset}}^{S} \inf(D_{s-1})\cdot w_{\max(J_{s-1})+1}\cdot\max(J_{s-1})\\
&=\sum_{\substack{s=1\\J_s\neq\emptyset}}^{S} \inf(D_s)\cdot w_{\min(J_{\geq s})}\cdot\Big(\min(J_{\geq s})-1\Big)\\
&\quad-\sum_{\substack{s=1\\J_{s-1}\neq\emptyset}}^{S} \inf(D_{s-1})\cdot w_{\min(J_{\geq s})}\cdot\Big(\min(J_{\geq s})-1\Big)\\
&\overset{(\triangle)}{=}\sum_{\substack{s=1\\J_{s-1}\neq\emptyset\\J_s\neq\emptyset}}^{S}\Big(\inf(D_s)-\inf(D_{s-1})\Big)\cdot w_{\min(J_{\geq s})}\cdot\Big(\min(J_{\geq s})-1\Big)\\
&\quad+\sum_{(s_1,s_2)\in\mathcal{E}}\Bigg(\inf(D_{s_2})\cdot w_{\min(J_{\geq s_2})}\cdot\Big(\min(J_{\geq s_2})-1\Big)\\
&\qquad\qquad-\inf(D_{s_1})\cdot w_{\min(J_{\geq s_1+1})}\cdot\Big(\min(J_{\geq s_1+1})-1\Big)\Bigg)\\
&\leq\sum_{(s_1,s_2)\in\mathcal{E}}\Bigg(\inf(D_{s_2})\cdot w_{\min(J_{\geq s_2})}\cdot\Big(\min(J_{\geq s_2})-1\Big)\\
&\qquad\qquad-\inf(D_{s_1})\cdot w_{\min(J_{\geq s_2})}\cdot\Big(\min(J_{\geq s_2})-1\Big)\Bigg)\\
&=\sum_{(s_1,s_2)\in\mathcal{E}}\Big(\inf(D_{s_2})-\inf(D_{s_1})\Big)\cdot w_{\min(J_{\geq s_2})}\cdot\Big(\min(J_{\geq s_2})-1\Big)\leq 0.
\end{aligned}
$$

The equality $(\triangle)$ is just a split of two sums into two cases: when two consecutive class C_{s-1}, C_s are non-empty or there is a positive number of empty classes between two non-empty classes C_{s_1}, C_{s_2}. □

For the second term from (2.15) we have

$$\begin{aligned}
&\sum_{\ell=1}^{n}\sum_{j\in\mathcal{C}}(w_\ell - w_{\ell+1})\cdot c_{\text{av}}^{T_\ell}(j)\\
=&\sum_{\ell=1}^{n}\sum_{j\in\mathcal{C}}\sum_{i\in\mathcal{F}}(w_\ell - w_{\ell+1})\cdot x_{ij}\cdot c_{ij}^{T_\ell}\\
=&\sum_{j\in\mathcal{C}}\sum_{i\in\mathcal{F}}x_{ij}\cdot\sum_{\ell=1}^{n}(w_\ell - w_{\ell+1})\cdot c_{ij}^{T_\ell}\\
\overset{(2.6)}{=}&\sum_{j\in\mathcal{C}}\sum_{i\in\mathcal{F}}x_{ij}\cdot\sum_{\substack{\ell=1\\ \ell:\ c_{ij}>T_\ell}}^{n}(w_\ell - w_{\ell+1})\cdot c_{ij}\\
=&\sum_{s=0}^{S}\sum_{\substack{j\in\mathcal{C},i\in\mathcal{F}\\ c_{ij}\in C_s}}x_{ij}\cdot c_{ij}\cdot\sum_{\substack{\ell=1\\ \ell\in J_{\geq s+1}}}^{n}(w_\ell - w_{\ell+1})\\
=&\sum_{s=0}^{S}\sum_{\substack{j\in\mathcal{C},i\in\mathcal{F}\\ c_{ij}\in C_s}}x_{ij}\cdot c_{ij}\cdot w_{\min\{J_{\geq s+1}\}}\\
\leq&\sum_{s=0}^{S}\sum_{\substack{j\in\mathcal{C},i\in\mathcal{F}\\ c_{ij}\in C_s}}x_{ij}\cdot c_{ij}\cdot w_{\text{av}}^{s}\leq(1+\epsilon)\sum_{s=0}^{S}\sum_{\substack{j\in\mathcal{C},i\in\mathcal{F}\\ c_{ij}\in C_s}}x_{ij}\cdot c_{ij}\cdot w_{\text{gs}}^{s}\\
=&(1+\epsilon)\sum_{j\in\mathcal{C}}\sum_{i\in\mathcal{F}}x_{ij}\cdot c_{ij}^{\text{r}}\overset{(2.1)}{=}(1+\epsilon)\cdot\text{OPT}^{*}.
\end{aligned}\tag{2.18}$$

This we can upper bound in terms of value of an optimal solution OPT. For that let us define the optimal solution W_{OPT} as a feasible solution of $LP(c^{\text{r}})$ and denote it as $(x^{\text{OPT}}, y^{\text{OPT}})$. It means that $y_i^{\text{OPT}}=1 \iff i\in W_{\text{OPT}}$ and $y_i^{\text{OPT}}=0 \iff i\notin W_{\text{OPT}}$.

$$\begin{aligned}
\text{OPT}^{*}\leq&\sum_{j\in\mathcal{C}}\sum_{i\in\mathcal{F}}x_{ij}^{\text{OPT}}c_{ij}^{\text{r}}=\sum_{j\in\mathcal{C}}\sum_{i\in\mathcal{F}}x_{ij}^{\text{OPT}}c_{ij}w(c_{ij})\\
=&\sum_{s=0}^{S}\sum_{\substack{j\in\mathcal{C},i\in\mathcal{F}\\ c_{ij}\in C_s}}x_{ij}^{\text{OPT}}c_{ij}w_{\text{gs}}^{s}\leq\sum_{s=0}^{S}\left(\max(D_s)\cdot w_{\text{gs}}^{s}\sum_{\substack{j\in\mathcal{C},i\in\mathcal{F}\\ c_{ij}\in C_s}}x_{ij}^{\text{OPT}}\right)\\
=&\sum_{s=0}^{S}\max(D_s)\cdot w_{\text{gs}}^{s}\cdot|C_s|\leq(1+\epsilon)^2\cdot\sum_{s=0}^{S}\inf(D_s)\cdot w_{\text{av}}^{s}\cdot|C_s|\\
=&(1+\epsilon)^2\cdot\sum_{s=0}^{S}\sum_{\ell\in J_s}w_\ell\cdot\inf(D_s)\leq(1+\epsilon)^2\cdot\text{OPT}.
\end{aligned}\tag{2.19}$$

In the end we have

$$\mathbb{E}\left[\text{cost}(A)\right]\overset{(2.15),(2.16),(2.17),(2.18),(2.19)}{\leq}(1+\epsilon)^3\cdot 38\cdot\text{OPT}.$$

Chapter 3

Harmonic k-Median and OWA k-Median

In the first part of this chapter we demonstrate how to use the Binary Negative Association property (BNA) of the Dependent Rounding procedure (DR) to derive a 2.36-approximation algorithm for the Harmonic k-Median problem (Theorem 23). First, we provide a detailed discussion on DR and BNA (Section 3.1), and several examples (Section 3.2). In Section 3.3 we provide a theorem and lemmas useful in our analysis.

In the second part of the chapter (Section 3.5) we show how to extend the 93-approximation algorithm of Hajiaghayi et al. [78] for Fault Tolerant k-Median into a 93-approximation algorithm for OWA k-Median in a metric space (Theorem 29).

Additional notation for this chapter.

For a natural number $t \in \mathbb{N}$, by $[t]$ we denote the set $\{1, 2, \ldots, t\}$. In this chapter we simplify notation for w^k (a vector of k weights) by writing just w.

3.1 Dependent Rounding and Negative Association

Consider a vector of m variables $(y_i)_{i\in[m]}$, and let y_i^* denote the initial value of the variable y_i. For simplicity we will assume that $0 \leq y_i^* \leq 1$ for each i, and that $k = \sum_{i\in[m]} y_i^*$ is an integer. A rounding procedure takes this vector of (fractional) variables as an input, and transforms it into a vector of 0/1 integers. We focus on a specific rounding procedure studied by Srinivasan [139] which we refer to as *dependent rounding* (DR).

DR works in steps: in each step it selects two fractional variables, say y_i and y_j, and changes the values of these variables to y_i' and y_j' so that $y_i' + y_j' = y_i + y_j$, and so that y_i' or y_j' is an integer. Thus, after each iteration at least one additional variable becomes an integer. The rounding procedure stops, when all variables are integers. In each step the randomization is involved: with some probability p variable

y_i is rounded to an integer value, and with probability $1-p$ variable y_j becomes an integer. The value of the probability p is selected so as to preserve the expected value of each individual entry y_i. Clearly, if $y_i + y_j \geq 1$, then one of the variables is rounded to 1; otherwise, one of the variables is rounded to 0. For example, if $y_i = 0.4$ and $y_j = 0.8$, then with probability 0.25 the values of the variables y_i and y_j change to, respectively, 1 and 0.2; and with probability 0.75 they change to, respectively, 0.2 and 1. If $y_i = 0.3$ and $y_j = 0.2$, then with probability 0.4 the values of the two variables change to, respectively, 0 and 0.5; and with probability 0.6, to, respectively, 0.5 and 0.

Let Y_i denote the random variable which returns one if y_i is rounded to one after the whole rounding procedure, and zero, otherwise. It was shown [139] that the DR generates distributions of Y_i which satisfy the following three properties:

Marginals. $\Pr[Y_i = 1] = y_i^*$,

Sum Preservation. $\Pr[\sum_i Y_i = k] = 1$,

Negative Correlation. For each $S \subseteq [m]$ it holds that
$\Pr[\bigwedge_{i\in S}(Y_i = 1)] \leq \prod_{i\in S} \Pr[Y_i = 1]$, and $\Pr[\bigwedge_{i\in S}(Y_i = 0)] \leq \prod_{i\in S} \Pr[Y_i = 0]$.

These three properties are often used in the analysis of approximation algorithms based on dependent rounding for various optimization problems—see, e.g., [75]. In fact, DR satisfies an even stronger property than NC, called *conditional negative association* (CNA) [99], yet, to the best of our knowledge, this property has never been used before for analyzing algorithms based on the DR procedure.

For two random variables, X and Y, by $\operatorname{cov}[X, Y]$ we denote the covariance between X and Y. Recall that $\operatorname{cov}[X, Y] = \mathbb{E}[XY] - \mathbb{E}[X] \cdot \mathbb{E}[Y]$.

Negative Association [87]. For each $S, Q \subseteq [m]$ with $S \cap Q = \emptyset$, $s = |S|$, and $q = |Q|$, and each two nondecreasing functions, $f\colon [0,1]^s \to \mathbb{R}$ and $g\colon [0,1]^q \to \mathbb{R}$, it holds that:

$$\operatorname{cov}\Big[f(Y_i\colon i \in S), g(Y_i\colon i \in Q)\Big] \leq 0.$$

Conditional Negative Association. We say that the sequence of random variables $(Y_i)_{i\in[m]}$ satisfies the CNA property if the conditional variables $(Y_{[m]\setminus S}|Y_S = a)$ satisfy NA for any $S \subseteq [m]$ and any $a = (a_i)_{i\in S}$. For $S = \emptyset$, CNA is equivalent to NA. It was shown by Dubhashi et al. [63] that if one rounds the variables according to a predefined linear order over the variables $\succ$ (i.e., if one always chooses for rounding the two fractional variables which are earliest in $\succ$), then the resulting distribution satisfies CNA. Yet, the requirement of following a predefined linear order of variables is too restrictive for our needs. Then, Kramer et al. [99] showed that DR following a predefined order on pairs of variables that implements a tournament tree returns a distribution satisfying CNA.

In our analysis we will use a simpler version of the NA property, which nevertheless is expressive enough for our needs. We introduce the following property.

Binary Negative Association (BNA). For each $S, Q \subseteq [m]$ with $S \cap Q = \emptyset$, $s = |S|$, and $q = |Q|$, and each two nondecreasing functions, $f\colon \{0,1\}^s \to \{0,1\}$ and $g\colon \{0,1\}^q \to \{0,1\}$, we have:

$$\operatorname{cov}\Big[f(Y_i\colon i \in S), g(Y_i\colon i \in Q)\Big] \leq 0.$$

From the definitions it is easy to see that CNA $\implies$ NA $\implies$ BNA.

3.2 Binary Negative Association is Strictly Stronger than Negative Correlation

We now argue that BNA is a strictly stronger property than NC. First we show a straightforward inductive argument that BNA implies NC. Next we provide an example of a distribution that satisfies NC but not BNA. In fact, this distribution is generated by a not-careful-enough implementation of DR.

Lemma 18. *For two binary random variables X, and Y, $X, Y \in \{0,1\}$, the condition* $\operatorname{cov}[X,Y] \leq 0$ *is equivalent to* $\Pr[X = 1 \wedge Y = 1] \leq \Pr[X = 1] \cdot \Pr[Y = 1]$.

Proof. Observe that for binary variables, X and Y, it holds that $\mathbb{E}[X] = \Pr[X = 1]$, $\mathbb{E}[Y] = \Pr[Y = 1]$, and $\mathbb{E}[XY] = \Pr[X = 1 \wedge Y = 1]$. □

Lemma 19. *Binary Negative Association of $(Y_i)_{i\in[m]}$ implies their Negative Correlation.*

Proof. We will prove the NC property by induction on $|S|$. Clearly, the property holds for $|S| = 1$. For an inductive step, we define two non-decreasing functions $f(Y_i\colon i \in S/\{j\}) = \bigwedge_{i\in S/\{j\}}(Y_i = 1)$ and $g(Y_j) = (Y_j = 1)$ for any $j \in S$.

$$\begin{aligned}
\Pr\left[\bigwedge_{i\in S}(Y_i = 1)\right] &= \Pr\left[\bigwedge_{i\in S/\{j\}}(Y_i = 1) \ \wedge\ Y_j = 1\right] \\
&\overset{\text{BNA, Lemma 18}}{\leq} \Pr\left[\bigwedge_{i\in S/\{j\}}(Y_i = 1)\right] \cdot \Pr\left[Y_j = 1\right] \\
&\overset{\text{inductive assum.}}{\leq} \prod_{i\in S}\Pr[Y_i = 1].
\end{aligned}$$

In order to bound the probability of $\bigwedge_{i\in S}(Y_i = 0)$ we define two other non-decreasing functions $f(Y_i\colon i \in S/\{j\}) = \bigvee_{i\in S/\{j\}}(Y_i > 0)$ and $g(Y_j) = (Y_j > 0)$

for any $j \in S$.

$$\begin{aligned}
\Pr\left[\wedge_{i\in S}(Y_i = 0)\right] &= 1 - \Pr\left[\bigvee_{i\in S}(Y_i > 0)\right] \\
&= 1 - \left(\Pr\left[\bigvee_{i\in S/\{j\}}(Y_i > 0)\right] + \Pr\left[Y_j > 0\right] - \Pr\left[\bigvee_{i\in S/\{j\}}(Y_i > 0) \ \wedge \ Y_j > 0\right]\right) \\
&= \Pr\left[\bigwedge_{i\in S/\{j\}}(Y_i = 0)\right] - \Pr\left[Y_j > 0\right] + \Pr\left[\bigvee_{i\in S/\{j\}}(Y_i > 0) \ \wedge \ Y_j > 0\right] \\
&\overset{\text{BNA, Lemma 18}}{\leq} \Pr\left[\bigwedge_{i\in S/\{j\}}(Y_i = 0)\right] - \Pr\left[Y_j > 0\right] + \Pr\left[\bigvee_{i\in S/\{j\}}(Y_i > 0)\right] \cdot \Pr\left[Y_j > 0\right] \\
&= \Pr\left[\bigwedge_{i\in S/\{j\}}(Y_i = 0)\right] - \Pr\left[Y_j > 0\right] \cdot \Pr\left[\bigwedge_{i\in S/\{j\}}(Y_i = 0)\right] \\
&= \Pr\left[\bigwedge_{i\in S/\{j\}}(Y_i = 0)\right] \cdot \Pr\left[Y_j = 0\right] \overset{\text{inductive assum.}}{\leq} \prod_{i\in S} \Pr[Y_i = 0].
\end{aligned}$$

□

Note that the general formulation of DR does not specify how the pairs of fractional variables are selected. The proof in [139] that DR satisfies NC is independent of the method in which these pairs of fractional variables are selected. We will now show that, if these pairs are selected by an adaptive adversary who may take into account the way in which the previous pairs were rounded, then the BNA property may not hold (so, also neither NA nor CNA). Consider the following example.

Example 3. *Consider $m = 8, k = 4$, and the vector of variables $(y_i)_{i\in[8]}$, all with the same initial value $1/2$. Let $S = \{2, 3, 4\}$, $Q = \{5\}$, and:*

$$f(Y_2, Y_3, Y_4) = \begin{cases} 1 & \text{if } \ Y_2 + Y_3 + Y_4 \geq 2 \\ 0 & \text{otherwise} \end{cases} \qquad g(Y_5) = Y_5.$$

Let α and β denote the events that $Y_2+Y_3+Y_4 \geq 2$ and that $Y_5 = 1$, respectively. BNA would require that $\Pr[\alpha \wedge \beta] \leq \Pr[\alpha] \cdot \Pr[\beta]$. Consider DR procedure as depicted in the following diagram (the paired variables are enclosed in rounded rectangles). First, we pair variables y_1 with y_5 and y_2 with y_6. The way in which the remaining variables are paired depends on the result of rounding within pairs (y_1, y_5) and (y_2, y_6). If y_1 and y_2 are both rounded to the same integer, then we pair y_3 with y_7 and y_4 with y_8. Otherwise, we pair y_3 with y_4 and y_7 with y_8.

Note that according to DR each rounding decision is taken with the same probability (e.g., when we pair variables y_1 with y_5, then the probabilities of y_1 and y_5 rounded to one is the same). Thus, we observe that $\Pr[\alpha] = 1/2$, $\Pr[\beta] = 1/2$, but $\Pr[\alpha \wedge \beta] = 1/4 + 1/16$.

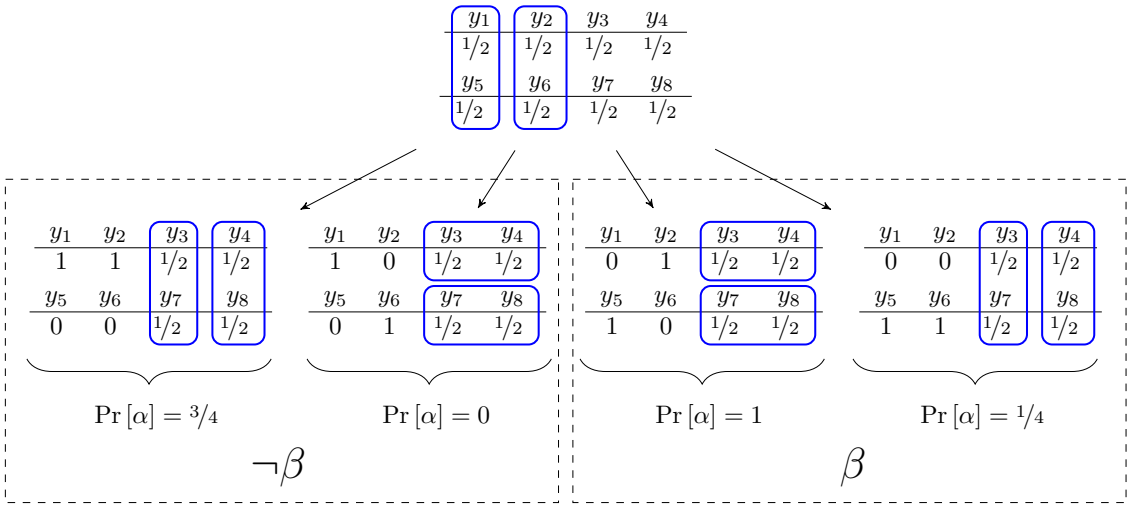


Figure 3.1: An illustration of Example 3.

Example 3 is simpler than the one given by Kramer et al. [99]. Both examples show that NA is a strictly stronger property than NC. Kramer et al. use the set of 7 variables with initial values equal to $3/7$, $k = 4$, and a predefined order on pairs of variables. Our example uses an adaptive adversary who decides which pair of variables should be rounded in each step of the rounding procedure. Our example cannot be implemented by fixing an order on pairs of variables (hence it also cannot be implemented by fixing a tournament tree). Our 8 variables have marginal probabilities equal to $1/2$, thus the example can be easily understood, and one does not need to calculate probabilities of choosing all $\binom{7}{4}$ 4-element sets.

3.3 Useful Lemmas

To make this thesis more self-contained we cite or prove a few useful inequalities.

Theorem 20 (Theorem 1.16 from [10])**.** *Let $X_1, X_2, \ldots, X_n$ be negatively correlated binary random variables. Let $X = \sum_{i=1}^{n} X_i$. Then X satisfies the Chernoff-Hoeffding bounds for $\delta \in [0, 1]$:*

$$\Pr\left[X \leq (1-\delta)\mathbb{E}[X]\right] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\mu}.$$

Lemma 21. *For any sequence $(a_i)_{i\in[n]}$ and $(b_i)_{i\in[n]}$, $b_i > 0$, it holds:*

$$\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} \leq \max_{i\in[n]} \frac{a_i}{b_i}.$$

Proof.

$$\begin{aligned}\frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} &= \sum_{j=1}^n \frac{a_j}{\sum_{i=1}^n b_i} = \sum_{j=1}^n \frac{a_j}{b_j} \cdot \frac{b_j}{\sum_{i=1}^n b_i} \leq \sum_{j=1}^n \left(\max_{i\in[n]} \frac{a_i}{b_i}\right) \frac{b_j}{\sum_{i=1}^n b_i} = \\ &= \left(\max_{i\in[n]} \frac{a_i}{b_i}\right) \sum_{j=1}^n \frac{b_j}{\sum_{i=1}^n b_i} = \max_{i\in[n]} \frac{a_i}{b_i}.\end{aligned}$$

□

Lemma 22. *For any non-decreasing sequence $(c_i)_{i\in[n]}$, $c_i > 0$ and any non-increasing sequence $(a_i)_{i\in[n]}$ it holds:*

$$\frac{\sum_{i=1}^n a_i c_i}{\sum_{i=1}^n c_i} \leq \frac{1}{n} \sum_{i=1}^n a_i.$$

Proof. We prove that by induction. Clearly, we have equality for $n = 1$. We assume that

$$\frac{\sum_{i=1}^{n-1} a_i c_i}{\sum_{i=1}^{n-1} c_i} \leq \frac{1}{n-1} \sum_{i=1}^{n-1} a_i.$$

It is equivalent to

$$(n-1) \cdot \sum_{i=1}^{n-1} a_i c_i \leq \left(\sum_{i=1}^{n-1} a_i\right) \cdot \left(\sum_{i=1}^{n-1} c_i\right). \tag{3.1}$$

We would like to show that

$$n \cdot \sum_{i=1}^{n} a_i c_i \leq \left(\sum_{i=1}^{n} a_i\right) \cdot \left(\sum_{i=1}^{n} c_i\right).$$

We have the following equivalent inequalities:

$$\begin{aligned}0 &\leq \left(\sum_{i=1}^{n-1} a_i\right) \cdot \left(\sum_{i=1}^{n-1} c_i\right) + a_n \cdot \sum_{i=1}^{n-1} c_i + c_n \cdot \sum_{i=1}^{n-1} a_i + a_n \cdot c_n - n \cdot \sum_{i=1}^{n-1} a_i c_i - n \cdot a_n \cdot c_n, \\ 0 &\leq \left[\left(\sum_{i=1}^{n-1} a_i\right) \cdot \left(\sum_{i=1}^{n-1} c_i\right) - (n-1) \cdot \sum_{i=1}^{n-1} a_i c_i\right] + \sum_{i=1}^{n-1} \left(a_n \cdot c_i + c_n \cdot a_i - a_n \cdot c_n - a_i \cdot c_i\right), \\ 0 &\leq \left[\left(\sum_{i=1}^{n-1} a_i\right) \cdot \left(\sum_{i=1}^{n-1} c_i\right) - (n-1) \cdot \sum_{i=1}^{n-1} a_i c_i\right] + \sum_{i=1}^{n-1} \left(a_i - a_n\right)\left(c_n - c_i\right).\end{aligned}$$

Using the inductive assumption (3.1) and monotonicity of sequences, i.e., $0 \leq a_i - a_n$, $0 \leq c_n - c_i$ we finish the proof. □

3.4 2.36-Approximation for Harmonic k-Median

In this section we construct and analyze a constant-factor approximation algorithm for Harmonic k-Median hence also for Proportional Approval Voting.

Theorem 23. *There exists a randomized algorithm for* Harmonic k-Median *that computes expected* 2.36*-approximate solution in polynomial time.*

Corollary 24. *There exists a randomized algorithm for the minimization* Proportional Approval Voting *that computes expected* 2.36*-approximate solution in polynomial time.*

In the remainder of this section we will prove the statement of Theorem 23. Consider the following linear program (2.1–2.5) that is a relaxation of a natural ILP for Harmonic k-Median.

$$\min \quad \sum_{j\in\mathcal{C}}\sum_{\ell=1}^{k}\sum_{i\in\mathcal{F}} w_\ell \cdot x_{ij}^\ell \cdot c_{ij} \qquad (3.2)$$

$$\sum_{i\in\mathcal{F}} y_i = k \qquad (3.3)$$

$$\sum_{\ell=1}^{k} x_{ij}^\ell \le y_i \qquad \forall i \in \mathcal{F},\ j \in \mathcal{C} \qquad (3.4)$$

$$\sum_{i\in\mathcal{F}} x_{ij}^\ell \ge 1 \qquad \forall j \in \mathcal{C},\ \ell \in [k] \qquad (3.5)$$

$$y_i, x_{ij}^\ell \in [0,1] \qquad \forall i \in \mathcal{F},\ j \in \mathcal{C},\ \ell \in [k] \qquad (3.6)$$

The intuitive meaning of the variables and constraints of the above LP is as follows. Variable y_i denotes how much facility i is opened. Integral values 1 and 0 correspond to, respectively, opening and not opening the facility. Constraint (3.3) encodes opening exactly k facilities. Each client $j \in \mathcal{C}$ has to be assigned to each among k opened facilities with different weights. For that we copy each client k times: the ℓ-th copy of a client j is assigned to the ℓ-th closest to j open facility. Variable x_{ij}^ℓ denotes how much the ℓ-th copy of j is assigned to facility i. In an integral solution we have $x_{ij}^\ell \in \{0,1\}$, which means that the ℓ-th copy of a client can be either assigned or not to the respective facility. The objective function (3.2) encodes the cost of assigning all copies of all clients to the opened facilities, applying proper weights. Constraint (3.4) prevents an assignment of a copy of a client to a not-opened part of a facility. In an integer solution it also forces assigning different copies of a client to different facilities. Observe that, due to non-increasing weights w_ℓ, the objective (3.2) is smaller if an ℓ'-th copy of a client is assigned to a closer facility than an ℓ''-th copy, whenever $\ell' < \ell''$. Constraint (3.5) ensures that each copy of a client is served by some facility.

Just like in most facility location settings it is crucial to select the facilities to open, and the later assignment of clients to facilities can be done optimally by a simple greedy procedure. We propose to select the set of facilities in a randomized way by applying the DR procedure to the y vector from an optimal fractional solution to linear program (2.1–2.5). This turns out to be a surprisingly effective methodology for Harmonic k-Median.

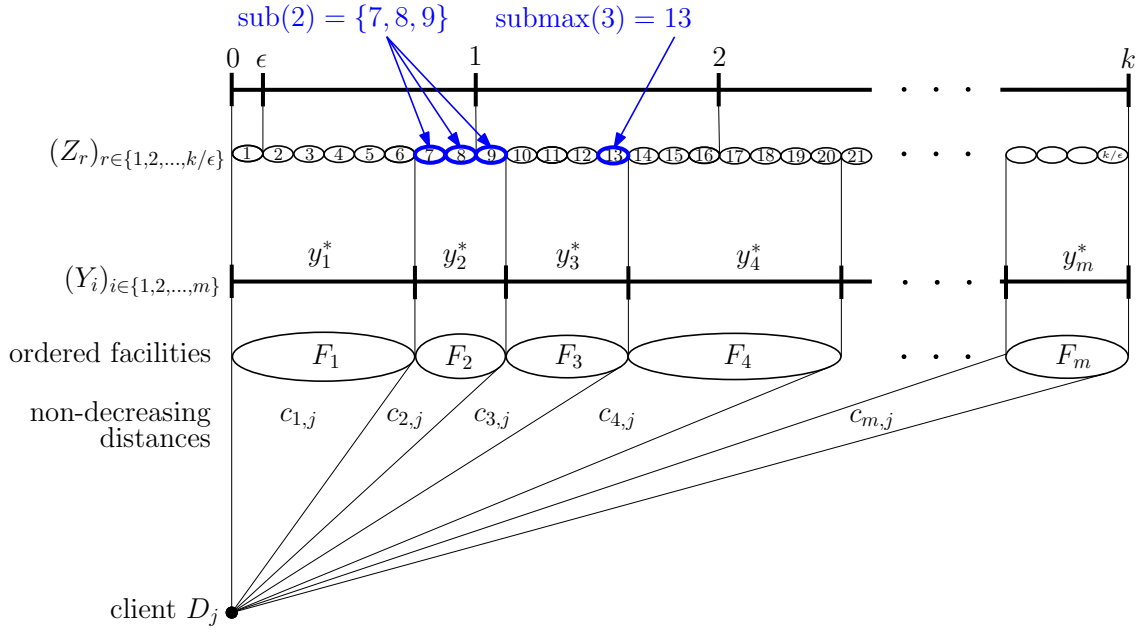


Figure 3.2: Ordering of the facilities by $c_{i,j}$ for the chosen client j. Pictorial definitions of the variables Y_i, Z_r and examples of the indices $\mathrm{sub}(i)$ and $\mathrm{submax}(i)$.

Analysis of the Algorithm

Let OPT^* be the value of an optimal solution (x^*, y^*) to the linear program (2.1–2.5). Let OPT be the value of an optimal solution $(x^{\mathrm{OPT}}, y^{\mathrm{OPT}})$ for Harmonic k-Median. Easily we can see that $(x^{\mathrm{OPT}}, y^{\mathrm{OPT}})$ is a feasible solution to the linear program (2.1–2.5), so $\mathrm{OPT}^* \leq \mathrm{OPT}$. Let $Y = (Y_1, \ldots, Y_m)$ be the random solution obtained by applying the DR procedure described in Section 3.1 to the vector y^*. Recall that DR preserves the sum of entries (see Section 3.1), hence we have exactly k facilities opened. It is straightforward to assign clients to the open facilities, so the variables $X = (X_{ij}^{\ell})_{j\in\mathcal{C},i\in\mathcal{F},\ell\in[k]}$ are easily determined.

We will show that the expected cost of solution Y denoted by $\mathbb{E}[\mathrm{cost}(Y)]$ is smaller than $2.36 \cdot \mathrm{OPT}^*$. In fact, we will show that $\mathbb{E}[\mathrm{cost}_j(Y)] < 2.36 \cdot \mathrm{OPT}^*_j$, where the subindex j extracts the cost of assigning client j to the facilities in the solution returned by the algorithm. In our analysis we focus on a single client $j \in \mathcal{C}$. Next, we reorder the facilities in the non-decreasing order of their connection costs to j (i.e., in the non-decreasing order of c_{ij}). Thus, from now on, facility F_i is the i-th closest facility to client j; ties are resolved in an arbitrary but fixed way.

The ordering of the facilities is depicted in Figure 3.2, which also includes information about the fractional opening of facilities in y^*, i.e., facility F_i is represented by an interval of length y_i^*. The total length of all intervals equals k. Next, we subdivide each interval into a set of (small) ϵ-size pieces (called ϵ-subintervals); ϵ is selected so that $1/\epsilon$, and y_i^*/ϵ for each i, are integers. Note that the values y_i^*, which originate from the solution returned by an LP solver, are rational numbers. The subdivision of $[0, k]$ into ϵ-subintervals is shown in Figure 3.2 on the "$(Z_r)_{r\in\{1,2,\ldots,k/\epsilon\}}$" level.

The idea behind introducing the ϵ-subintervals is the following. Although compu-

tationally the algorithm applies DR to the y^* variables, for the sake of the analysis we may think that the DR process is actually rounding z variables corresponding to ϵ-subinterval under the additional assumption that rounding within individual facilities is done before rounding between facilities. Formally, we replace the vector $Y = (Y_1, Y_2, \ldots, Y_m)$ by an equivalent vector of random variables $Z = (Z_1, Z_2, \ldots, Z_{k/\epsilon})$. Random variable Z_r represents the r-th ϵ-subinterval. We will use the following notation to describe the bundles of ϵ-subintervals that correspond to particular facilities:

$$\text{submax}(0) = 0 \qquad \text{and} \qquad \text{submax}(i) = \text{submax}(i-1) + \frac{y_i^*}{\epsilon}, \tag{3.7}$$

$$\text{sub}(i) = \{\text{submax}(i-1) + 1, \ldots, \text{submax}(i)\}. \tag{3.8}$$

Intuitively, $\text{sub}(i)$ is the set of indexes r such that Z_r represents an interval belonging to the i-th facility. Examples for both definitions are shown in Figure 3.2 in the upper level. Formally, the random variables Z_r are defined so that:

$$Y_i = \sum_{r \in \text{sub}(i)} Z_r \qquad \text{and} \qquad Y_i = 1 \implies \exists!\, r \in \text{sub}(i) \quad Z_r = 1. \tag{3.9}$$

For each $r \in \{1, 2, \ldots, k/\epsilon\}$ we can write that:

$$\Pr[Z_r = 1] = \Pr[Z_r = 1 \,|\, Y_{\text{sub}^{-1}(r)} = 1] \cdot \Pr[Y_{\text{sub}^{-1}(r)} = 1] = \frac{\epsilon}{y^*_{\text{sub}^{-1}(r)}} \cdot y^*_{\text{sub}^{-1}(r)} = \epsilon \tag{3.10}$$

and $\Pr[Z_r = 0] = 1 - \epsilon$, hence $\mathbb{E}[Z_r] = \epsilon$. Also we have:

$$\Pr[Y_i = 1] = \Pr\left[\sum_{r \in \text{sub}(i)} Z_r = 1\right] = \Pr\left[\bigvee_{r \in \text{sub}(i)} Z_r = 1\right] = \sum_{r \in \text{sub}(i)} \Pr[Z_r = 1]. \tag{3.11}$$

When $Y_i = 1$ its representative is chosen randomly among $(Z_r)_{r \in \text{sub}(i)}$ independently of the choices of representatives of other facilities. Therefore

$$\forall_{i \in [m]} \quad \forall_{r \in \text{sub}(i)} \quad \mathbb{E}[f(Y) \mid Y_i = 1] = \mathbb{E}[f(Y) \mid Y_i = 1 \ \wedge \ Z_r = 1], \tag{3.12}$$

for any function f on vector $Y = (Y_1, Y_2, \ldots, Y_m)$.

Now we are ready to analyze the expected cost for any client $j \in \mathcal{C}$.

$$
\begin{aligned}
\mathbb{E}[\mathrm{cost}_j(Y)] &\leq \sum_{i=1}^{m}\left(\mathbb{E}\left[\frac{c_{ij}}{1+\sum_{i'=1}^{i-1}Y_{i'}}\middle|Y_i=1\right]\cdot\Pr\left[Y_i=1\right]\right)\\
&\overset{(3.11)}{=} \sum_{i=1}^{m}\left(c_{ij}\cdot\mathbb{E}\left[\frac{1}{1+\sum_{i'=1}^{i-1}Y_{i'}}\middle|Y_i=1\right]\cdot\sum_{r\in\mathrm{sub}(i)}\Pr\left[Z_r=1\right]\right)\\
&= \sum_{i=1}^{m}\left(c_{ij}\cdot\sum_{r\in\mathrm{sub}(i)}\mathbb{E}\left[\frac{1}{1+\sum_{i'=1}^{i-1}Y_{i'}}\middle|Y_i=1\right]\cdot\Pr\left[Z_r=1\right]\right)\\
&\overset{(3.12)}{=} \sum_{i=1}^{m}\left(c_{ij}\cdot\sum_{r\in\mathrm{sub}(i)}\mathbb{E}\left[\frac{1}{1+\sum_{i'=1}^{i-1}Y_{i'}}\middle|Y_i=1\wedge Z_r=1\right]\cdot\Pr\left[Z_r=1\right]\right)\\
&\overset{(3.9),(3.10)}{=} \sum_{i=1}^{m}\left(\epsilon\cdot c_{ij}\cdot\sum_{r\in\mathrm{sub}(i)}\mathbb{E}\left[\frac{1}{1+\sum_{r'=1}^{\mathrm{submax}(i-1)}Z_{r'}}\middle|Z_r=1\right]\right)\\
&\overset{(3.9)}{=} \sum_{i=1}^{m}\left(\epsilon\cdot c_{ij}\cdot\sum_{r\in\mathrm{sub}(i)}\mathbb{E}\left[\frac{1}{1+\sum_{r'=1}^{r-1}Z_{r'}}\middle|Z_r=1\right]\right) \qquad (3.13)
\end{aligned}
$$

W.l.o.g., assume that $\mathrm{OPT}_j^* > 0$. Hence the approximation ratio for any client j is

$$
\frac{\mathbb{E}[\mathrm{cost}_j(Y)]}{\mathrm{OPT}_j^*} \overset{(3.8),(3.13)}{\leq} \frac{\sum_{r=1}^{k/\epsilon}\epsilon\cdot c_{\mathrm{sub}^{-1}(r),j}\cdot\mathbb{E}\left[\frac{1}{1+\sum_{r'=1}^{r-1}Z_{r'}}\middle|Z_r=1\right]}{\sum_{r=1}^{k/\epsilon}\epsilon\cdot c_{\mathrm{sub}^{-1}(r),j}\cdot\frac{1}{\lceil r\epsilon\rceil}} =
$$

note that $\mathrm{sub}^{-1}(r)$ is an index of a facility that contains Z_r. Now we convert the sum over facilities into a sum over unit intervals. A unit interval is represented as a sum of $1/\epsilon$ many ϵ-subintervals:

$$
= \frac{\sum_{\ell=1}^{k}\sum_{r=(\ell-1)/\epsilon+1}^{\ell/\epsilon} c_{\mathrm{sub}^{-1}(r),j}\cdot\mathbb{E}\left[\frac{1}{1+\sum_{r'=1}^{r-1}Z_{r'}}\middle|Z_r=1\right]}{\sum_{\ell=1}^{k}\sum_{r=(\ell-1)/\epsilon+1}^{\ell/\epsilon} c_{\mathrm{sub}^{-1}(r),j}\cdot\frac{1}{\ell}} \leq
$$

W.l.o.g., we can assume that first interval has non-zero costs: $\sum_{r=1}^{1/\epsilon} c_{\mathrm{sub}^{-1}(r),j} > 0$, otherwise the LP pays 0 and our algorithm pays 0 in expectation on intervals from non-empty prefix of $(1,2,\ldots,k)$. With this assumption we can take maximum over intervals:

$$
\overset{\text{Lemma 21}}{\leq} \max_{\ell\in[k]}\left(\frac{\sum_{r=(\ell-1)/\epsilon+1}^{\ell/\epsilon} c_{\mathrm{sub}^{-1}(r),j}\cdot\mathbb{E}\left[\frac{1}{1+\sum_{r'=1}^{r-1}Z_{r'}}\middle|Z_r=1\right]}{\sum_{r=(\ell-1)/\epsilon+1}^{\ell/\epsilon} c_{\mathrm{sub}^{-1}(r),j}\cdot\frac{1}{\ell}}\right) \leq
$$

Costs $c_{\text{sub}^{-1}(r),j}$ can be general and they could be hard to analyze. Therefore we would like to remove costs from the analysis. We will use Lemma 22 for which the technique of splitting variables Y_i into Z_r was needed. We are using the fact that the variables Z_r have the same expected values; otherwise the coefficient in front of the expected value would be $c_{ij} \cdot y_i^*$, i.e., not monotonic. Thus

$$\overset{\text{Lemma 22}}{\leq} \max_{\ell \in [k]} \left(\epsilon \cdot \ell \cdot \sum_{r=(\ell-1)/\epsilon+1}^{\ell/\epsilon} \mathbb{E}\left[\frac{1}{1+\sum_{r'=1}^{r-1} Z_{r'}} \middle| Z_r = 1 \right] \right). \tag{3.14}$$

Consider the expected value in the above expression for a fixed $r \in \{{}^{(\ell-1)}/_{\epsilon}+1, \ldots, {}^{\ell}/_{\epsilon}\}$:

$$E_r = \mathbb{E}\left[\frac{1}{1+\sum_{r'=1}^{r-1} Z_{r'}} \middle| Z_r = 1 \right] = \sum_{t=1}^{k} \frac{1}{t} \Pr\left[\sum_{r'=1}^{r-1} Z_{r'} = t-1 \middle| Z_r = 1 \right] =$$
$$= \sum_{t=1}^{\ell} \frac{1}{t} \Pr\left[\sum_{r'=1}^{r-1} Z_{r'} = t-1 \middle| Z_r = 1 \right] + \sum_{t=\ell+1}^{k} \frac{1}{t} \Pr\left[\sum_{r'=1}^{r-1} Z_{r'} = t-1 \middle| Z_r = 1 \right]. \tag{3.15}$$

For $t \in \{1, 2, \ldots, \ell\}$ we consider the conditional probability in the above expression, denote it by $p_r(t-1)$, and analyze the corresponding cumulative distribution function $H_r(t-1)$:

$$p_r(t-1) = \Pr\left[\sum_{r'=1}^{r-1} Z_{r'} = t-1 \middle| Z_r = 1 \right], \tag{3.16}$$
$$H_r(t-1) = \Pr\left[\sum_{r'=1}^{r-1} Z_{r'} \leq t-1 \middle| Z_r = 1 \right] = \sum_{t'=0}^{t-1} p_r(t'), \tag{3.17}$$

for which we have the following recursive relation

$$\begin{aligned} p_r(0) &= H_r(0), \\ p_r(t-1) &= H_r(t-1) - H_r(t-2) \quad \forall_{t \in \{2,3,\ldots,\ell\}}. \end{aligned}$$

We continue the analysis of E_r:

$$\begin{aligned}
E_r &\overset{(3.15),(3.16)}{=} \sum_{t=1}^{\ell}\frac{1}{t}p_r(t-1)+\sum_{t=\ell+1}^{k}\frac{1}{t}p_r(t-1)\\
&\overset{(3.17)}{=} H_r(0)+\sum_{t=2}^{\ell}\frac{1}{t}\left(H_r(t-1)-H_r(t-2)\right)+\sum_{t=\ell+1}^{k}\frac{1}{t}p_r(t-1)\\
&= H_r(0)+\sum_{t=2}^{\ell}\frac{1}{t}H_r(t-1)-\sum_{t=2}^{\ell}\frac{1}{t}H_r(t-2)+\sum_{t=\ell+1}^{k}\frac{1}{t}p_r(t-1)\\
&= \sum_{t=1}^{\ell}\frac{1}{t}H_r(t-1)-\sum_{t=1}^{\ell-1}\frac{1}{t+1}H_r(t-1)+\sum_{t=\ell+1}^{k}\frac{1}{t}p_r(t-1)\\
&= \sum_{t=1}^{\ell-1}\frac{1}{t}H_r(t-1)-\sum_{t=1}^{\ell-1}\frac{1}{t+1}H_r(t-1)+\frac{1}{\ell}H_r(\ell-1)+\sum_{t=l+1}^{k}\frac{1}{t}p_r(t-1)\\
&\leq \sum_{t=1}^{\ell-1}\left(\frac{1}{t}-\frac{1}{t+1}\right)H_r(t-1)+\frac{1}{\ell}\left(H_r(\ell-1)+\sum_{t=\ell+1}^{k}p_r(t-1)\right)\\
&= \sum_{t=1}^{\ell-1}\frac{1}{t(t+1)}H_r(t-1)+\frac{1}{\ell}\left(H_r(\ell-1)+\sum_{t=\ell+1}^{k}p_r(t-1)\right)\\
&\leq \sum_{t=1}^{\ell-1}\frac{1}{t(t+1)}H_r(t-1)+\frac{1}{\ell}. \qquad (3.18)
\end{aligned}$$

We will use the following two lemmas to remove the conditioning and bound $H_r(t-1)$.

Lemma 25. *Distribution of $\{Z_1, Z_2, \ldots, Z_{k/\epsilon}\}$ satisfies Binary Negative Association.*

Proof. Note that DR procedure on $(Y_i)_{i\in[m]}$ and then independent choice of $(Z_r)_{r\in\mathrm{sub}(i)}$ for each $i\in[m]$ is equivalent to the following implementation of DR on $(Z_r)_{r\in\{1,2,\ldots,k/\epsilon\}}$. First, for each $i\in[m]$ $(Z_r)_{r\in\mathrm{sub}(i)}$ are processed until obtaining a single non-zero variable that is equivalent to y_i. Then, in the second phase the rounding proceeds as if it had started from the y_i variables. Since this process altogether is an implementation of a single DR procedure with fixed tournament tree starting from $(Z_r)_{r\in\{1,2,\ldots,k/\epsilon\}}$ variables, we can simply apply the result of Kramer et al. [99] and get the statement of the lemma.

At this point we note that the result of Dubhashi et al. [63] is not sufficient for proving our lemma. They have proved that DR following a predefined order of variables (which can be viewed as a linear tournament tree) returns distributions satisfying the CNA property. Here, however, we need to have at least a "two-stage" linear tournament: the first linear tournament on variables $(Z_r)_{r\in\mathrm{sub}(i)}$ and the second tournament on winning variables from the first tournament. □

In the following lemma we combine the BNA property of variables $\{Z_1, Z_2, \ldots, Z_{k/\epsilon}\}$ with applications of Chernoff-Hoeffding bounds

Lemma 26. *For any $\ell \in [k]$, $t \in [\ell - 1]$ and $r \in \{{}^{(\ell-1)}/_{\epsilon} + 1, {}^{(\ell-1)}/_{\epsilon} + 2, \ldots, {}^{\ell}/_{\epsilon}\}$ we have*

$$H_r(t-1) \leq e^{-r\cdot\epsilon} \cdot \left(\frac{e \cdot r \cdot \epsilon}{t}\right)^t.$$

Proof. Let us fix $\ell \in [k]$, $t \in [\ell - 1]$ and $r \in \{{}^{(\ell-1)}/_{\epsilon} + 1, {}^{(\ell-1)}/_{\epsilon} + 2, \ldots, {}^{\ell}/_{\epsilon}\}$. We have

$$\begin{aligned} H_r(t-1) &\overset{(3.17)}{=} \Pr\left[\sum_{r'=1}^{r-1} Z_{r'} \leq t-1 \;\middle|\; Z_r = 1\right] \\ &= \Pr\left[\sum_{r'=r}^{k/\epsilon} Z_{r'} \geq k-(t-1) \;\middle|\; Z_r = 1\right] \\ &= \Pr\left[\sum_{r'=r+1}^{k/\epsilon} Z_{r'} \geq k-t \;\middle|\; Z_r = 1\right]. \end{aligned} \tag{3.19}$$

We now exploit Binary Negative Association of variables Z_i (Lemma 25). By setting $S = \{r+1, r+2, \ldots, {}^{k}/_{\epsilon}\}, Q = \{r\}, f(a_1, a_2, \ldots a_s) = \mathbb{1}\left\{\sum_{i=1}^{|S|} a_i \geq k-t\right\}$ and $g(a) = a$ we obtain:

$$0 \geq \operatorname{cov}\Big[f(Z_{r'} : r' \in S), g(Z_{r'} : r' \in Q)\Big] = \operatorname{cov}\left[\mathbb{1}\left\{\sum_{r'=r+1}^{k/\epsilon} Z_{r'} \geq k-t\right\}, Z_r\right].$$

Since f, g are binary and non-decreasing we can use Lemma 18 to obtain an equivalent inequality:

$$\Pr\left[\sum_{r'=r+1}^{k/\epsilon} Z_{r'} \geq k-t \quad\wedge\quad Z_r = 1\right] \leq \Pr\left[\sum_{r'=r+1}^{k/\epsilon} Z_{r'} \geq k-t\right] \cdot \Pr\left[Z_r = 1\right]. \tag{3.20}$$

Therefore,

$$\begin{aligned} H_r(t-1) &\overset{(3.19)}{\leq} \Pr\left[\sum_{r'=r+1}^{k/\epsilon} Z_{r'} \geq k-t \;\middle|\; Z_r = 1\right] \\ &= \frac{\Pr\left[\sum_{r'=r+1}^{k/\epsilon} Z_{r'} \geq k-t \quad\wedge\quad Z_r = 1\right]}{\Pr\left[Z_r = 1\right]} \\ &\overset{(3.20)}{\leq} \Pr\left[\sum_{r'=r+1}^{k/\epsilon} Z_{r'} \geq k-t\right] = \Pr\left[\sum_{r'=1}^{r} Z_{r'} \leq t\right]. \end{aligned} \tag{3.21}$$

Using Lemma 25 and Lemma 19 we know that $(Z_r)_{r\in\{1,2,\ldots,k/\epsilon\}}$ are negatively correlated. What is more, t is smaller than the expected value of the sum

$$t \leq \ell - 1 = (\ell - 1 + \epsilon) - \epsilon \leq r\cdot\epsilon - \epsilon < r\cdot\epsilon \overset{(3.10)}{=} \mathbb{E}\left[\sum_{r'=1}^{r} Z_{r'}\right],$$

Therefore, we can use Chernoff-Hoeffding bounds as follows

$$\begin{aligned} H_r(t-1) &\overset{(3.21)}{\leq} \Pr\left[\sum_{r'=1}^{r} Z_{r'} \leq t\right] \\ &= \Pr\left[\sum_{r'=1}^{r} Z_{r'} < r\cdot\epsilon\cdot\left(1-\left(1-\frac{t}{r\cdot\epsilon}\right)\right)\right] \\ &\overset{\text{Theorem 20}}{\leq} \left(\frac{e^{\frac{t}{r\cdot\epsilon}-1}}{\left(\frac{t}{r\cdot\epsilon}\right)^{\frac{t}{r\cdot\epsilon}}}\right)^{r\cdot\epsilon} = \frac{e^{t-r\cdot\epsilon}\cdot(r\cdot\epsilon)^t}{t^t} \\ &= e^{-r\cdot\epsilon}\cdot\left(\frac{e\cdot r\cdot\epsilon}{t}\right)^t. \end{aligned}$$

□

We are ready to show an approximation ration for any client $j \in \mathcal{C}$.

$$\begin{aligned} \frac{\mathbb{E}[\text{cost}_j(Y)]}{\text{OPT}_j^*} &\overset{(3.14),(3.15),(3.18)}{\leq} \max_{\ell\in[k]} \epsilon\cdot\ell\cdot\sum_{r=(\ell-1)/\epsilon+1}^{\ell/\epsilon}\left(\sum_{t=1}^{\ell-1}\left(\frac{1}{t(t+1)}\cdot H_r(t-1)\right)+\frac{1}{\ell}\right) \\ &= 1+\max_{\ell\in[k]} \epsilon\cdot\ell\cdot\sum_{r=(\ell-1)/\epsilon+1}^{\ell/\epsilon}\left(\sum_{t=1}^{\ell-1}\frac{1}{t(t+1)}\cdot H_r(t-1)\right) \\ &\overset{\text{Lemma 26}}{\leq} 1+\max_{\ell\in[k]} \epsilon\cdot\ell\cdot\sum_{r=(\ell-1)/\epsilon+1}^{\ell/\epsilon}\left(\sum_{t=1}^{\ell-1}\frac{1}{t(t+1)}\cdot e^{-r\cdot\epsilon}\cdot\left(\frac{e\cdot r\cdot\epsilon}{t}\right)^t\right) \\ &= 1+\max_{\ell\in[k]} \ell\cdot\sum_{t=1}^{\ell-1}\frac{1}{t(t+1)}\cdot\frac{e^t}{t^t}\cdot\left(\sum_{r=(\ell-1)/\epsilon+1}^{\ell/\epsilon}\epsilon\cdot e^{-r\cdot\epsilon}\cdot(r\cdot\epsilon)^t\right) \\ &= 1+\max_{\ell\in[k]} \ell\cdot\sum_{t=1}^{\ell-1}\frac{1}{t(t+1)}\cdot\frac{e^t}{t^t}\cdot\left(\sum_{r=(\ell-1)/\epsilon+1}^{\ell/\epsilon-1}\int_{r\cdot\epsilon-\epsilon}^{r\cdot\epsilon} e^{-r\cdot\epsilon}\cdot(r\cdot\epsilon)^t\,dx\right) \end{aligned}$$

we now use an upper bound on the most interior sum by an integral of the function $f_t(x) = e^{-x}\cdot x^t$. Note that $f_t'(x) = e^{-x}\cdot x^{t-1}\cdot(t-x) \leq 0$ for $1 \leq t \leq \ell-1 \leq x$, so the function f is non-increasing. Therefore

$$\leq 1+\max_{\ell\in[k]} \ell\cdot\sum_{t=1}^{\ell-1}\frac{1}{t(t+1)}\cdot\frac{e^t}{t^t}\cdot\left(\int_{\ell-1}^{\ell} e^{-x}\cdot x^t dx\right). \tag{3.22}$$

To bound the above expression we first numerically evaluate it for $\ell \in \{1,2,\ldots,88\}$ and obtain

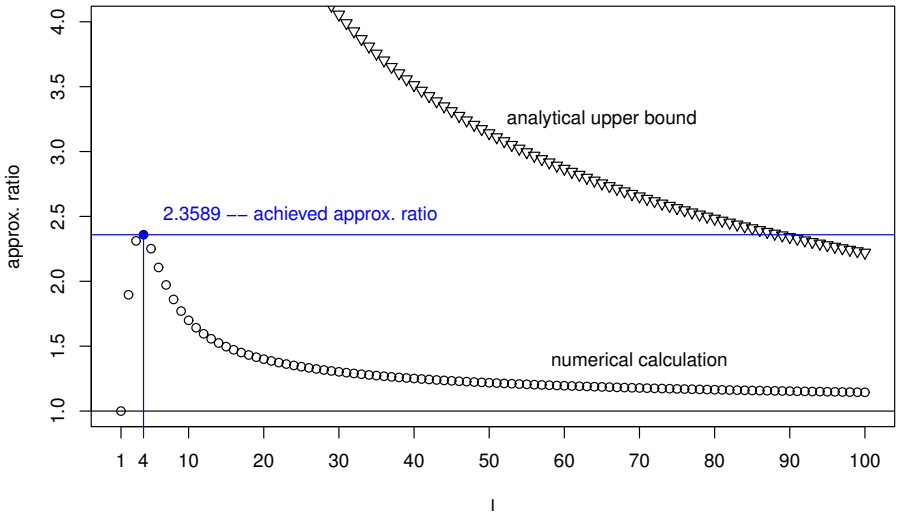
$$1+\max_{\ell\in\{1,2,\ldots,88\}} \ell\cdot\sum_{t=1}^{\ell-1}\frac{1}{t(t+1)}\cdot\frac{e^t}{t^t}\cdot\left(\int_{\ell-1}^{\ell} e^{-x}\cdot x^t dx\right) < 2.3589 < 2.36.$$

It remains to bound the expression for $\ell \in \{89,90,\ldots,k\}$, which we do by the fol-

lowing estimation:

$$\begin{aligned}
&1+\ell\cdot\sum_{t=1}^{\ell-1}\frac{1}{t(t+1)}\cdot\frac{e^t}{t^t}\cdot\left(\int_{\ell-1}^{\ell}e^{-x}\cdot x^t dx\right)\\
&\quad\leq\quad 1+\ell\cdot\sum_{t=1}^{\ell-1}\frac{1}{t(t+1)}\cdot\frac{e^t}{t^t}\cdot e^{-(\ell-1)}\cdot\ell^t\\
&\overset{\text{Stirling}}{\leq}\quad 1+\ell\cdot\sum_{t=1}^{\ell-1}\frac{1}{t(t+1)}\cdot\frac{\sqrt{2\pi t}\cdot e^{\frac{1}{12t}}}{t!}\cdot e^{-(\ell-1)}\cdot\ell^t\\
&\quad\leq\quad 1+\sqrt{2\pi}\cdot e^{\frac{1}{12}}\cdot e^{-(\ell-1)}\cdot\frac{1}{\sqrt{\ell}}\cdot\sum_{t=1}^{\ell-1}\frac{\ell^{t+1}}{(t+1)!}\cdot\frac{\sqrt{\ell}}{\sqrt{t}}\\
&\quad\leq\quad 1+\sqrt{2\pi}\cdot e^{\frac{13}{12}}\cdot e^{-\ell}\cdot\frac{1}{\sqrt{\ell}}\cdot\sum_{t=1}^{\ell-1}\frac{\ell^{t+1}}{(t+1)!}\cdot\frac{\ell}{t}\\
&\quad\leq\quad 1+3\sqrt{2\pi}\cdot e^{\frac{13}{12}}\cdot e^{-\ell}\cdot\frac{1}{\sqrt{\ell}}\cdot\sum_{t=1}^{\ell-1}\frac{\ell^{t+1}}{(t+1)!}\cdot\frac{\ell}{t+2}\\
&\overset{\text{Taylor series for } e^\ell}{\leq}\quad 1+3\sqrt{2\pi}\cdot e^{\frac{13}{12}}\cdot e^{-\ell}\cdot\frac{1}{\sqrt{\ell}}\cdot e^{\ell}\\
&\quad=\quad 1+3\sqrt{2\pi}\cdot e^{\frac{13}{12}}\cdot\frac{1}{\sqrt{\ell}}<2.3551<2.36.
\end{aligned}$$

The maximum is obtained for $\ell=4$ (see Figure 3.3).



ℓ	**ratio**
1	1
2	1.90
3	2.32
4	**2.36**
5	2.26
6	2.11
7	1.98
8	1.86
9	1.78
10	1.70
11	1.65
12	1.60

Figure 3.3: The numerical and the analytical upper bound on the approximation ratio on intervals $(\ell-1,\ell)$, for each $\ell\in[k]$.

3.5 OWA k-MEDIAN with the Triangle Inequality

In this section we construct an algorithm for OWA k-MEDIAN with costs satisfying the triangle inequality. Thus, the problem we address in this section is more general than HARMONIC k-MEDIAN (i.e., the problem we have considered in the previous section) in a sense that we allow for arbitrary non-increasing sequences of weights. On the other hand, it is less general in a sense that we require the costs to form a specific structure (a metric).

In our approach we first adapt the algorithm of Hajiaghayi et al. [78] for FAULT TOLERANT k-MEDIAN so that it applies to the following, slightly more general setting: for each client j we introduce its multiplicity $m_j \in \mathbb{N}$—intuitively, this corresponds to cloning j and co-locating all such clones in the same location as j. However, this will require a modification of the original algorithm for FAULT TOLERANT k-MEDIAN, since we want to allow the multiplicities $\{m_j\}_{j\in\mathcal{C}}$ to be exponential with respect to the size of the instance (otherwise, we could simply copy each client a sufficient number of times, and use the original algorithm of Hajiaghayi et al.).

Next, we provide a reduction from OWA k-MEDIAN to such a generalization of FAULT TOLERANT k-MEDIAN. The resulting FAULT TOLERANT k-MEDIAN WITH CLIENTS MULTIPLICITIES problem can be cast as the following integer program:

$$\begin{aligned}
\min \quad & \sum_{j\in\mathcal{C}}\sum_{i\in\mathcal{F}} m_j \cdot x_{ij} \cdot c_{ij} && \sum_{i\in\mathcal{F}} x_{ij} = r_j && \forall j \in \mathcal{C} \\
& \sum_{i\in\mathcal{F}} y_i = k && x_{ij} \leq y_i && \forall i \in \mathcal{F},\ j \in \mathcal{C} \\
& && y_i, x_{ij} \in \{0,1\} && \forall i \in \mathcal{F} \\
& && m_j \in \mathbb{N} && \forall j \in \mathcal{C}
\end{aligned}$$

In the subsequent theorem we show that the 93-approximation algorithm of Hajiaghayi et al. can be generalized so that it applies to FAULT TOLERANT k-MEDIAN WITH CLIENTS MULTIPLICITIES with costs forming a metric.

Theorem 27. *There exists a randomized algorithm for* METRIC FAULT TOLERANT k-MEDIAN WITH CLIENTS MULTIPLICITIES *that computes expected* 93*-approximate solution in polynomial time.*

Proof. We reduce an instance of FAULT TOLERANT k-MEDIAN WITH CLIENTS MULTIPLICITIES to an instance of FAULT TOLERANT k-MEDIAN by replacing the multiple m_j of a client j with m_j clients in the same location and with the same connectivity requirement (we will call such clients clones of j). Observe that there exists an optimal solution in which each clone of the same client is connected to the same set of open facilities. Next, we run the 93-approximation algorithm of Hajiaghayi et al. [78] on such a constructed instance with clones. It is apparent that the solution that we obtain by following this procedure approximates the original instance with the ratio of 93. However, the issue is that m_j can be exponential in the number of clients in the original instance, and so the most straightforward implementation of our reduction does not run in polynomial-time. To deal with that we will efficiently encode the

reduced instance, and we will show that the algorithm of Hajiaghayi et al. can be adapted to run on such encoded instances. We proceed as follows.

First, we solve the LP part of the original algorithm [78] with the additional multiplicative factors $\{m_j\}_{j\in\mathcal{C}}$ added to the objective function. From the solution to the LP, $(y_i)_{i\in\mathcal{F}}$ with $\sum_{i\in\mathcal{F}} y_i = k$, we construct an optimal assignment of the clients to the facilities. We encode such an assignment efficiently by grouping all clones of the same client into a single cluster and storing an assignment for a single client for each cluster only (we call such a client the *representative* of the cluster). In particular, note that all clones in the same cluster have the same assignments and so, they all have the same average and maximal assignment costs. We use this property in the next step of the original algorithm: *creating bundles* of volume 1 [78, Algorithm 1]. By a careful analysis of this algorithm we can observe that no new bundles are created for a cloned client (lines 5 and 6 of Algorithm 1 in [78]) and so that the cloned clients can be considered in bunches.

Next, as in the original algorithm, we divide the clients into *safe* and *dangerous* by the criterion on the ratio of the maximal and the average cost in the assignment vector. Intuitively, if the maximum is much higher than the average then the client is marked as dangerous (for a formal definition see Section 2.2 in [78]), otherwise it is considered safe. Hence, the clones of the same client are either all safe or all dangerous. In the latter case they are also *in conflict*: they are close and they have the same connectivity requirements (for a definition also see Section 2.2 in [78]). Thus, in the *filtering phase* [78, Algorithm 2] either all the dangerous clones of the same client are filtered out or exactly one of them survives; without loss of generality we can assume that the representative of the cluster survives. In fact, this is the main reason why we can quite easy adapt the algorithm. The next step, that is building a laminar family [78, Algorithm 3], is independent on clients that were filtered out, and so it can be performed on our efficiently encoded instance. The safe clients are not used later on by the algorithm (they are only the side effect of creating bundles and later on they only appear in the algorithm's analysis). Finally, the rounding process of the algorithm [78, Section 2.3] depends on the set of constructed bundles and on the set of filtered dangerous clients (and the induced laminar family), and as we discussed it is possible to construct each of the two families with efficient encoding. This completes the proof. □

Consider reduction from OWA k-Median to Fault Tolerant k-Median with Clients Multiplicities depicted on Figure 3.4.

Lemma 28. *Let I be an instance of* OWA k-Median*, and let I' be an instance of* Fault Tolerant k-Median with Clients Multiplicities *constructed from I through reduction from Figure 3.4. An α-approximate solution to I' is also an α-approximate solution to I.*

Proof. Let $\mathcal{W}$ be an α-approximate solution to I'. By FT-k-med-multi$(\mathcal{W}, j)$ we denote be the total cost of the clients $j_1, j_2, \ldots, j_k$ constructed through reduction from Figure 3.4. Similarly, let OWA-k-med$(\mathcal{W}, j)$ be the cost of the client j for a

Reduction. *Let us take an instance I of* OWA k-MEDIAN $(\mathcal{C}, \mathcal{F}, k, w, \{c_{ij}\}_{i\in\mathcal{F}, j\in\mathcal{C}})$ *where $w_i = \frac{p_i}{q_i}, i \in [k]$ are rational numbers in the canonical form. We construct an instance I' of* FAULT TOLERANT k-MEDIAN WITH CLIENTS MULTIPLICITIES *with the same set of facilities and the same number of facilities to open, k. Each client $j \in \mathcal{C}$ is replaced with clients $j_1, j_2, \ldots, j_k$ with requirements $1, 2, \ldots, k$, respectively. For $Q = \prod_{r=1}^{k} q_r$, the multiples of the clients are defined as follows:*

- $m_{j_\ell} = (w_\ell - w_{\ell+1}) \cdot Q$, *for each* $\ell \in [k-1]$, *and*
- $m_{j_k} = w_k \cdot Q$.

Figure 3.4: Reduction from OWA k-MEDIAN to FAULT TOLERANT k-MEDIAN WITH CLIENTS MULTIPLICITIES.

solution $\mathcal{W}$ in I. For each client j we have:

$$\begin{aligned}
\text{FT-k-med-multi}(\mathcal{W}, j) &= \sum_{r=1}^{k} m_{j_r} \cdot \left(\sum_{i=1}^{r} c_i^{\rightarrow}(\mathcal{W}, j) \right) = \sum_{r=1}^{k} \sum_{i=1}^{r} m_{j_r} \cdot c_i^{\rightarrow}(\mathcal{W}, j) \\
&= \sum_{r=1}^{k} \sum_{i=1}^{r} m_{j_r} \cdot c_i^{\rightarrow}(\mathcal{W}, j) = \sum_{i=1}^{k} \sum_{r=i}^{k} m_{j_r} \cdot c_i^{\rightarrow}(\mathcal{W}, j) \\
&= \sum_{i=1}^{k} \left(c_i^{\rightarrow}(\mathcal{W}, j) \cdot \sum_{r=i}^{k} m_{j_r} \right) \\
&= \sum_{i=1}^{k} c_i^{\rightarrow}(\mathcal{W}, j) \cdot w_i \cdot Q = Q \cdot \text{OWA-k-med}(\mathcal{W}, j).
\end{aligned}$$

Let $\mathcal{W}_I^*$ and $\mathcal{W}_{I'}^*$ be optimal solutions for I and I', respectively. By the same reasoning, we have that:

$$\text{FT-k-med-multi}(\mathcal{W}_I^*, j) = Q \cdot \text{OWA-k-med}(\mathcal{W}_I^*, j).$$

And, thus, that:

$$\begin{aligned}
&\sum_{j\in\mathcal{C}} \text{OWA-k-med}(\mathcal{W}, j) = \sum_{j\in\mathcal{C}} \frac{1}{Q} \text{FT-k-med-multi}(\mathcal{W}, j) \\
&\leq \alpha \frac{1}{Q} \sum_{j\in\mathcal{C}} \text{FT-k-med-multi}(\mathcal{W}_{I'}^*, j) \leq \alpha \frac{1}{Q} \sum_{j\in\mathcal{C}} \text{FT-k-med-multi}(\mathcal{W}_I^*, j) \\
&= \alpha \sum_{j\in\mathcal{C}} \text{OWA-k-med}(\mathcal{W}_I^*, j).
\end{aligned}$$

This completes the proof. □

Since our reduction preserves the structure of the costs, we immediately obtain an approximation algorithm for the metric variant of our problem.

Theorem 29. *There exists a randomized algorithm for* Metric OWA k-Median *that computes expected* 93*-approximate solution in polynomial time.*

Proof. We use the reduction from Figure 3.4 and Theorem 27. The approximation ratio follows from Lemma 28. □

Chapter 4

Minimax Approval Voting

In this chapter we provide three algorithms for the Minimax Approval Voting problem. In Section 4.1 we show the first polynomial time approximation scheme (PTAS) for Minimax Approval Voting (Theorem 39) which improves the previous best 2-approximation due to Caragiannis et al. [39]. Next, in Section 4.2 we construct a parameterized approximation scheme (Theorem 40) which we use in Section 4.3 to show a faster PTAS (Theorem 46).

Additional notation for this chapter.

For a natural number $t \in \mathbb{N}$, by $[t]$ we denote the set $\{1, 2, \ldots, t\}$. For a string $s \in \{0,1\}^m$, the number of 1's in s is denoted as $n_1(s)$ and it is also called the Hamming weight of s; similarly $n_0(s) = m - n_1(s)$ denotes the number of zeroes. Moreover, the set of all strings of length m with k ones is denoted by $S_{k,m}$, i.e., $S_{k,m} = \{s \in \{0,1\}^m : n_1(s) = k\}$. $s[j]$ for $j \in [m]$ means the j-th letter of a string s. $s_i[j] = 1$ if voter i approves candidate j-th and $s_i[j] = 0$ if voter i does not approve candidate j-th. For a subset of positions $P \subseteq [m]$ we define a subsequence $s|_P$ by removing the letters at positions $[m] \setminus P$ from s.

We generalize the Hamming distance for real-valued strings $x, y \in [0,1]^m$ by $\mathcal{H}(x,y) = \sum_{j=1}^{m} \left|x[j] - y[j]\right|$. For a set of words $S \subseteq \{0,1\}^m$ and a word $x \in \{0,1\}^m$ we denote $\mathcal{H}(x,S) = \max_{s \in S} \mathcal{H}(x,s)$.

For a string $s \in \{0,1\}^m$, any string $s' \in S_{k,m}$ at distance $|n_1(s) - k|$ from s is called a *k-completion* of s. Note that it is easy to find such a k-completion s': when $n_1(s) \geq k$ we obtain s' by replacing arbitrary $n_1(s) - k$ ones in s by zeroes; similarly when $n_1(s) < k$ we obtain s' by replacing arbitrary $k - n_1(s)$ zeroes in s by ones. In this chapter we assume k-completion is made by a deterministic rule.

Throughout this chapter OPT denotes the value of an optimal solution s_{OPT} for the given instance $(\{s_i\}_{i \in [n]}, k)$ of Minimax Approval Voting, i.e., $\text{OPT} = \max_{i \in [n]} \mathcal{H}(s_{\text{OPT}}, s_i)$.

We will use the following Chernoff-Hoeffding bounds.

Theorem 30. *[118, chapter 4.1] Let $X_1, X_2, \ldots, X_n$ be n independent binary random variables such that for every $i \in [n]$ we have $\Pr[X_i = 1] = p_i$, for $p_i \in [0, 1]$. Let $X = \sum_{i=1}^{n} X_i$. Then,*

- *for any $0 < \epsilon \leq 1$ we have:*

$$\Pr[X > (1+\epsilon) \cdot \mathbb{E}[X]] \leq \exp\left(-\tfrac{1}{3}\epsilon^2 \cdot \mathbb{E}[X]\right) \tag{4.1}$$

$$\Pr[X < (1-\epsilon) \cdot \mathbb{E}[X]] \leq \exp\left(-\tfrac{1}{2}\epsilon^2 \cdot \mathbb{E}[X]\right) \tag{4.2}$$

- *for any $1 < \epsilon$ we have:*

$$\Pr[X > (1+\epsilon) \cdot \mathbb{E}[X]] \leq \exp\left(-\tfrac{1}{3}\epsilon \cdot \mathbb{E}[X]\right) \tag{4.3}$$

$$\Pr[X < (1-\epsilon) \cdot \mathbb{E}[X]] = 0 \tag{4.4}$$

4.1 The First Polynomial Time Approximation Scheme

In this section we will show the first polynomial time approximation scheme for Minimax Approval Voting. First, in Subsection 4.1.1 we formalize the information we may extract from subset of votes, and introduce a measure of inaccuracy of such a subset. Next, in Subsection 4.1.2 we prove the existence of a small subset of votes with stable inaccuracy. In Subsection 4.1.3 we show that the optimization problem of deciding the part of the committee not induced by the subset of votes can be approximated with only a small additional loss in the objective function. Finally, in Subsection 4.1.4 we give an algorithm considering all subsets of a fixed size and show that, in the iteration when the algorithm happens to consider a subset with stable inaccuracy, it will produce a $(1+\epsilon)$-approximate solution to Minimax Approval Voting.

4.1.1 Extracting Information from Subsets

We consider subsets of votes and analyze the information they carry. We measure the inaccuracy of this information with respect to the set of all votes. We show that there exists a small subset with stable inaccuracy, i.e., the drop of inaccuracy after including one more vote is small.

Let us define an inaccuracy function $\mathrm{ina} : 2^S \mapsto \mathbb{N}_{\geq 0}$ that measures the inaccuracy if we will consider subset $Y \subseteq S$ instead of S. The smaller the $\mathrm{ina}(Y)$ is the better the common parts of strings in Y represent s_{OPT}.

Definition 31. *For all $Y \subseteq S, Y \neq \emptyset$ we define $t_Y \in \{0,1\}^m$ and* $\mathrm{ina}(Y)$ *as:*

$$t_Y[j] = \begin{cases} 0 & \text{if } \forall_{y \in Y} \quad y[j] = 0, \\ 1 & \text{if } \forall_{y \in Y} \quad y[j] = 1, \\ s_{\mathrm{OPT}}[j] & \text{otherwise.} \end{cases}$$

$$\mathrm{ina}(Y) = \mathcal{H}(t_Y, s_{\mathrm{OPT}}).$$

Intuitively t_Y is the optimal solution s_{OPT} changed at positions where all strings from Y agree. Also we define the pattern of a subset of votes.

Definition 32. *For all $Y \subseteq S, Y \neq \emptyset$ we define pattern $p_Y \in \{0,1,*\}^m$ as:*

$$p_Y[j] = \begin{cases} 0 & \text{if } \forall_{y \in Y} \quad y[j] = 0, \\ 1 & \text{if } \forall_{y \in Y} \quad y[j] = 1, \\ * & \text{otherwise.} \end{cases}$$

It represents positions that all strings in Y agree. "$*$" encodes a mismatch. Note that (from Definitions 31 and 32) t_Y is an optimal solution s_{OPT} overwritten by a pattern p_Y on no-star positions:

$$t_Y[j] = \begin{cases} s_{\mathrm{OPT}}[j] & \text{if } p_Y[j] = * \\ p_Y[j] & \text{otherwise.} \end{cases}$$

The inaccuracy function has the following property:

Lemma 33. $\forall_{s_{i_1} \in S}$, *for all sequences $\{s_{i_1}\} = Y_1 \subseteq Y_2 \subseteq \cdots \subseteq Y_n = S$ we have*

$$\mathrm{OPT} \geq \mathrm{ina}(Y_1) \geq \mathrm{ina}(Y_2) \geq \cdots \geq \mathrm{ina}(Y_n) = 0.$$

Proof. It is easy to see that

$$\mathrm{ina}(Y_1) \stackrel{\text{def.}}{=} \mathcal{H}(t_{Y_1}, s_{\mathrm{OPT}}) = \mathcal{H}(t_{\{s_{i_1}\}}, s_{\mathrm{OPT}}) = \mathcal{H}(s_{i_1}, s_{\mathrm{OPT}}) \leq \mathrm{OPT},$$

$$\mathrm{ina}(Y_n) = \mathrm{ina}(S) = \mathcal{H}(s_{\mathrm{OPT}}, s_{\mathrm{OPT}}) = 0.$$

Still we need to prove $\mathrm{ina}(Y_i) \geq \mathrm{ina}(Y_{i+1})$. Pattern $p_{Y_{i+1}}$ is built on strings from $Y_i \subseteq Y_{i+1}$ and strings from $Y_{i+1} \setminus Y_i$. So $p_{Y_{i+1}}$ has at least as many $*$'s as p_{Y_i} has. Therefore $t_{Y_{i+1}}$ has at least as many positions as t_{Y_i} has that agree with optimal solution s_{OPT}, so $\mathcal{H}\left(t_{Y_i}, s_{\mathrm{OPT}}\right) \geq \mathcal{H}\left(t_{Y_{i+1}}, s_{\mathrm{OPT}}\right)$. Using definition of the inaccuracy function (Definition 31) we prove the lemma. □

Intuitively $\mathrm{ina}(Y) - \mathrm{ina}(Y \cup \{y\})$ is the decrease of the inaccuracy from adding element y to set Y. We will show that, when adding one more element y to sets Y, Z such that $Y \subseteq Z$, the inaccuracy decrease more in a case of adding y to the smaller set Y than adding y to the bigger set Z.

Lemma 34. *If we artificially extend the* $\mathrm{ina}(\cdot)$ *function for the empty set as* $\mathrm{ina}(\emptyset) = 2 \cdot \mathrm{OPT}$*, then the* $\mathrm{ina}(\cdot)$ *function is supermodular*[1]*, i.e.,*

$$\forall_{Y \subseteq Z \subseteq S} \quad \forall_{s \in S} \quad \mathrm{ina}(Z) - \mathrm{ina}(Z \cup \{s\}) \leq \mathrm{ina}(Y) - \mathrm{ina}(Y \cup \{s\}) \tag{4.5}$$

Proof. Let fix Y, Z and s such that $Y \subseteq Z \subseteq S$ and $s \in S$.

Case 1: $Z = \emptyset$:
Then also $Y = \emptyset$, and inequality (4.5) holds obviously.

Case 2: $Z \neq \emptyset, Y = \emptyset$:
We have:

$$\begin{aligned}\mathrm{ina}(Z) - \mathrm{ina}(Z \cup \{s\}) \leq \mathrm{OPT} = 2 \cdot \mathrm{OPT} - \mathrm{OPT} \leq \mathrm{ina}(\emptyset) - \mathrm{ina}(\{s\}) \\ = \mathrm{ina}(Y) - \mathrm{ina}(Y \cup \{s\}),\end{aligned}$$

because we use respectively: Lemma 33 and the fact that Z has at least one element; definition of $\mathrm{ina}(\cdot)$ for empty set and an upper bound for $\mathrm{ina}(\cdot)$ function; assumption that $Y = \emptyset$.

Case 3: $Z \neq \emptyset, Y \neq \emptyset$:
From definition of $\mathrm{ina}(\cdot)$ we have:

$$\mathrm{ina}(Z) - \mathrm{ina}(Z \cup \{s\}) = \mathcal{H}\Big(t_Z, s_{\mathrm{OPT}}\Big) - \mathcal{H}\Big(t_{Z \cup \{s\}}, s_{\mathrm{OPT}}\Big)$$

counting a difference by considering two cases for value of s_{OPT} we obtain

$$\begin{aligned} = \quad & \Big|\Big\{j : s_{\mathrm{OPT}}[j] = 1 \wedge t_{Z \cup \{s\}}[j] = 1 \wedge t_Z[j] = 0\Big\}\Big| \\ + & \Big|\Big\{j : s_{\mathrm{OPT}}[j] = 0 \wedge t_{Z \cup \{s\}}[j] = 0 \wedge t_Z[j] = 1\Big\}\Big| \end{aligned}$$

using definition of $t_{(\cdot)}$ we get

$$\begin{aligned} = \quad & \Big|\Big\{j : s_{\mathrm{OPT}}[j] = 1 \wedge s[j] = 1 \wedge \ \forall_{z \in Z}\ z[j] = 0\Big\}\Big| \\ + & \Big|\Big\{j : s_{\mathrm{OPT}}[j] = 0 \wedge s[j] = 0 \wedge \ \forall_{z \in Z}\ z[j] = 1\Big\}\Big| \end{aligned}$$

taking an universal quantifier over a smaller subset we obtain

$$\begin{aligned} \leq \quad & \Big|\Big\{j : s_{\mathrm{OPT}}[j] = 1 \wedge s[j] = 1 \wedge \ \forall_{y \in Y}\ y[j] = 0\Big\}\Big| \\ + & \Big|\Big\{j : s_{\mathrm{OPT}}[j] = 0 \wedge s[j] = 0 \wedge \ \forall_{y \in Y}\ y[j] = 1\Big\}\Big| \end{aligned}$$

reversing all previous transformations finally we obtain

$$= \mathrm{ina}(Y) - \mathrm{ina}(Y \cup \{s\}).$$

□

[1] according to [134, page 766], $f : 2^S \mapsto \mathbb{R}$ is supermodular iff $\forall_{Y,Z \subseteq S} \quad f(Y) + f(Z) \leq f(Y \cup Z) + f(Y \cap Z)$ which is equivalent with $\forall_{Y \subseteq Z \subseteq S} \quad \forall_{s \in S} \quad f(Z) - f(Z \cup \{s\}) \leq f(Y) - f(Y \cup \{s\})$.

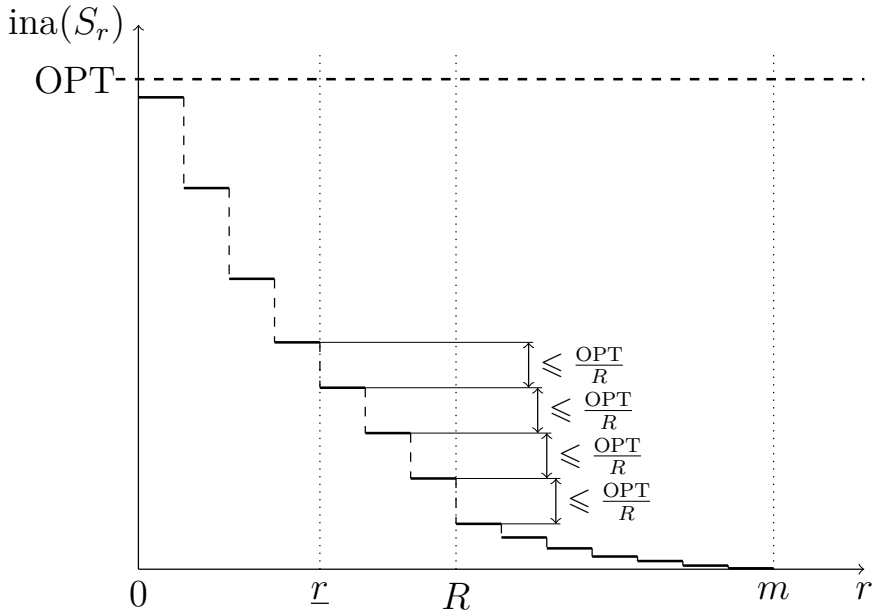


Figure 4.1: The inaccuracy function ina$(\cdot)$ for the sequence of subsets $S_1 \subset S_2 \subset \cdots \subset S_n = S$.

4.1.2 Existence of a Stable Subset

In this section we will show there exists R-size subset of votes X that when adding one more vote into X the inaccuracy decreases by at most $\frac{\text{OPT}}{R}$.

Lemma 35. *For any fixed $R \in \mathbb{N}_{\geq 1}$ there exists a subset $X \subseteq S, |X| = R$ such that*

$$\forall_{s \in S \setminus X} \quad \text{ina}(X) - \text{ina}(X \cup \{s\}) \leq \frac{\text{OPT}}{R}. \tag{4.6}$$

We say such X is $\frac{\text{OPT}}{R}$-stable.

Proof. First, we construct $S_{\underline{r}}$ satisfying (4.6) with at most R elements.

Let us construct a sequence of subsets $S_1 \subset S_2 \subset \cdots \subset S_n = S, |S_i| = i$. We take $S_1 = \{s_{i_1}\}$, where s_{i_1} is any element of S and for $r \in \{2, 3, \ldots, n\}$ we take $S_r = S_{r-1} \cup \{s_{i_r}\}$ where s_{i_r} is such a vote that after adding it the inaccuracy function decreases the most, i.e.,

$$s_{i_r} = \operatorname*{arg\,max}_{s \in S \setminus S_{r-1}} \Big(\text{ina}(S_{r-1}) - \text{ina}(S_{r-1} \cup \{s\})\Big). \tag{4.7}$$

We have

$$\min_{r \in \{1,2,\ldots,R\}} \text{ina}(S_r) - \text{ina}(S_{r+1}) \leq \frac{1}{R}\left(\sum_{r=1}^{R} \text{ina}(S_r) - \text{ina}(S_{r+1})\right) =$$

$$= \frac{1}{R}\Big(\text{ina}(S_1) - \text{ina}(S_{R+1})\Big) \leq \frac{\text{OPT}}{R}, \tag{4.8}$$

because (from Lemma 33) we know that $\mathrm{ina}(S_1) \leq \mathrm{OPT}$ and $\mathrm{ina}(S_{R+1}) \geq 0$. Let $\underline{r}$ be a minimizer for the left-hand side of (4.8), then (by the choice of $s_{i_{\underline{r}}}$ in (4.7)) we have:

$$\max_{s \in S \setminus S_{\underline{r}}} \Big(\mathrm{ina}(S_{\underline{r}}) - \mathrm{ina}(S_{\underline{r}} \cup \{s\}) \Big) \leq \frac{\mathrm{OPT}}{R}, \tag{4.9}$$

thus $S_{\underline{r}}$ satisfies (4.6), see Figure 1. If $S_{\underline{r}}$ has less elements than R we can extend $S_{\underline{r}}$ to an R-elements subset X by adding any elements of $S \setminus S_{\underline{r}}$. It follows from the supermodularity of $\mathrm{ina}(\cdot)$. From Lemma 34 we have:

$$\forall_{s \in S \setminus S_{\underline{r}}} \quad \mathrm{ina}(X) - \mathrm{ina}(X \cup \{s\}) \leq \mathrm{ina}(S_{\underline{r}}) - \mathrm{ina}(S_{\underline{r}} \cup \{s\}),$$

and hence also:

$$\max_{s \in S \setminus S_{\underline{r}}} \Big(\mathrm{ina}(X) - \mathrm{ina}(X \cup \{s\}) \Big) \leq \max_{s \in S \setminus S_{\underline{r}}} \Big(\mathrm{ina}(S_{\underline{r}}) - \mathrm{ina}(S_{\underline{r}} \cup \{s\}) \Big). \tag{4.10}$$

Finally, taking (4.9) and (4.10) we obtain:

$$\max_{s \in S \setminus X} \Big(\mathrm{ina}(X) - \mathrm{ina}(X \cup \{s\}) \Big) \leq \frac{\mathrm{OPT}}{R}.$$

□

Of course we cannot construct such a subset efficiently if we do not know s_{OPT}. How to find a proper subset X? For constructing our PTAS we will fix $R \in \mathbb{N}_{\geq 1}$ and consider all subsets $Y \subseteq S$ with cardinality R. There is less than $n^R \in \mathrm{poly}(n)$ such subsets. For clarity, we will use $Y \subseteq S$ in arguments valid for all subsets considered by the algorithm, and $X \subseteq S$ for a $\frac{\mathrm{OPT}}{R}$-stable subset of votes.

For a fixed $Y \subseteq S, Y \neq \emptyset$, w.l.o.g. we reorder candidates in such a way that p_Y is a lexicographically smallest permutation:

$$p_Y = * * \ldots * 00 \ldots 011 \ldots 1.$$

The first part (from the left) is called "star positions" or "star part". The remaining part is called "no-star part". We define $p^{(*)}(Y)$ as the number of $*$ in p_Y and we denote it β:

$$\beta = p_Y^{(*)} = \Big| \{ j : p_Y[j] = * \} \Big|.$$

In our PTAS we essentially fix the "no-star part" of the answer to the pattern p_Y and optimize over the choices for the "star part" of the outcome. If the number of stars or number of 1's on star positions of s_{OPT} is small enough, then there is only $\mathrm{poly}(n, m)$ possible solutions and we can consider all of them. Let us analyze the size of the "star part".

Lemma 36. *For all $Y \subseteq S$ we have*

$$\beta = p^{(*)}(Y) \leq |Y| \cdot \mathrm{OPT}$$

Proof. Consider an arbitrary $Y = \{y_1, y_2, \ldots, y_{|Y|}\}$. We can construct Y in the following 3 phases:

1. $Y := \{s_{\text{OPT}}\}$
2. for $i \in \{1, 2, \ldots, |Y|\}$ do
   $Y := Y \cup \{y_i\}$
3. $Y := Y \setminus \{s_{\text{OPT}}\}$

After that we obtain set Y. Let us calculate how many stars p_Y has. In Phase 1 there are no stars. In each step in Phase 2 we add at most OPT stars, because $\forall_{i\in\{1,2,\ldots,|Y|\}} \quad \mathcal{H}(y_i, s_{\text{OPT}}) \leq \text{OPT}$. In Phase 3 we can at most decrease the number of stars. So $\beta \leq |Y| \cdot \text{OPT}$. □

Note that for X from Lemma 35 we have

$$p^{(*)}(X) \leq |X| \cdot \text{OPT} = R \cdot \text{OPT}.$$

Let us now introduce some more notation. Assuming $Y \subseteq S$ and hence also $\beta = p^{(*)}(Y)$ are fixed, we will use the following notation to denote the "star part" and the "no-star part" of a string $x \in \{0,1\}^m$:

$$x' = x[1] \cdot x[2] \cdot \ldots \cdot x[\beta],$$

$$x'' = x[\beta+1] \cdot x[\beta+2] \cdot \ldots \cdot x[m],$$

where"$\cdot$" is a concatenation of strings (letters). So we divide x into two parts: $x = x' \cdot x''$.

In the following lemma we will show that for the pattern from a stable subset X we can change the number of 1's in the "no-star part" to the properly guessed number of 1's loosing only twice the stability constant.

Lemma 37. *If $X \subseteq S$ is $(\epsilon_1 \cdot \text{OPT})$-stable, z'' is a k''-completion of p''_X, where $k'' = n_1(s''_{\text{OPT}})$, then*

$$\forall_{i\in\{1,2,\ldots,n\}} \quad \mathcal{H}(s'_{\text{OPT}} \cdot z'', s_i) \leq (1 + 2\epsilon_1) \cdot \text{OPT}. \tag{4.11}$$

Proof. W.l.o.g. there is insufficient number of 1's in no-star part of pattern p_X, i.e., $k'' \geq n_1(p''_X)$. The other case is symmetric.

Let us fix $s_i \in S$ and consider all combinations of values in strings p''_X, z'', s''_i, s''_{OPT} at the same position j. $\alpha_a \in \mathbb{N}$, for $a \in \{1, 2, \ldots, 12\}$, counts the number of positions j with combination a, see Table 4.1.
We have

$$\mathcal{H}(z'', s''_i) = \Big|\{j : z''[j] \neq s''_i[j]\}\Big|$$

and we consider two cases for value of s_{OPT} at position j:

$$= \Big|\{j : z''[j] \neq s''_i[j] \wedge (z''[j] = s_{\text{OPT}} \vee z''[j] \neq s_{\text{OPT}})\}\Big|$$

next, we divide it into two components:

$$\begin{aligned} = \quad & \Big|\{j : s_{\text{OPT}} = z''[j] \neq s''_i[j] \qquad\}\Big| \\ + & \Big|\{j : \qquad\quad z''[j] \neq s''_i[j] = s_{\text{OPT}})\}\Big| \end{aligned}$$

	combinations											
index of a combination	1	2	3	4	5	6	7	8	9	10	11	12
$p''_X[j]$	0	0	0	0	0	0	0	0	1	1	1	1
$z''[j]$	0	0	0	0	1	1	1	1	1	1	1	1
$s''_i[j]$	0	1	0	1	0	1	0	1	0	1	0	1
$s''_{\mathrm{OPT}}[j]$	0	0	1	1	0	0	1	1	0	0	1	1
number of occurrences	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	α_{12}
$\mathcal{H}(z''[j], s''_i[j])$	0	1	0	1	1	0	1	0	1	0	1	0
$\mathcal{H}(s''_{\mathrm{OPT}}[j], s''_i[j])$	0	1	1	0	0	1	1	0	0	1	1	0

Table 4.1: Combinations of values in strings $p''_X, z'', s''_i, s''_{\mathrm{OPT}}$. There is only 12 combinations (no $2^4 = 16$), because by the assumption $k'' \geq n_1(p''_X)$ we never change from 1 in p''_X to 0 in z''.

we use case counts from Table 4.1 to count positions in both components:

$$= (\underbrace{\alpha_2 + \alpha_7 + \alpha_{11}}_{\text{first component}} + \underbrace{\alpha_3 + \alpha_6 + \alpha_{10} - \alpha_3 - \alpha_6 - \alpha_{10}}_{=0}) + \underbrace{(\alpha_4 + \alpha_5 + \alpha_9)}_{\text{second component}} =$$

and we use the definition of the Hamming distance:

$$= \Big(\mathcal{H}(s''_{\mathrm{OPT}}, s''_i) - \alpha_3 - \alpha_6 - \alpha_{10}\Big) + (\alpha_4 + \alpha_5 + \alpha_9). \tag{4.12}$$

Since $n_1(z'') = k'' = n_1\Big(s''_{\mathrm{OPT}}\Big)$, we get

$$\sum_{k=5}^{12} \alpha_k = \alpha_3 + \alpha_4 + \alpha_7 + \alpha_8 + \alpha_{11} + \alpha_{12}$$

$$\alpha_5 = \alpha_3 + \alpha_4 - \alpha_6 - \alpha_9 - \alpha_{10}. \tag{4.13}$$

Also we have

$$\alpha_4 + \alpha_8 + \alpha_9 \leq \epsilon_1 \cdot \mathrm{OPT}, \tag{4.14}$$

because X is $\epsilon_1 \cdot \mathrm{OPT}$-stable. Now we are ready to prove equation (4.11).

$$\begin{aligned}
\mathcal{H}(s'_{\mathrm{OPT}} \cdot z'', s_i) &\overset{\text{def.}}{=} \mathcal{H}(s'_{\mathrm{OPT}}, s'_i) + \mathcal{H}(z'', s''_i) \\
&\overset{(4.12)}{=} \mathcal{H}(s'_{\mathrm{OPT}}, s'_i) + \mathcal{H}(s''_{\mathrm{OPT}}, s''_i) - \alpha_3 - \alpha_6 - \alpha_{10} + \alpha_4 + \alpha_5 + \alpha_9 \\
&\overset{(4.13)}{=} \underbrace{\mathcal{H}(s_{\mathrm{OPT}}, s_i)}_{\leq \mathrm{OPT}} + 2\Big(\underbrace{\alpha_4}_{\overset{(4.14)}{\leq} \epsilon_1 \cdot \mathrm{OPT}} - \alpha_6 - \alpha_{10}\Big) \overset{(4.14)}{\leq} (1 + 2\epsilon_1) \cdot \mathrm{OPT}.
\end{aligned}$$

□

4.1.3 An Auxiliary Optimization Problem

In this section we will consider the optimization problem obtained after guessing the number of 1's in the two parts and fixing the "no-star part" of the outcome. It has variables for all the positions of the "star part" and constraints for all the original votes $s_i \in S$.

Let us define the optimization problem $\mathrm{IP}(Y, k')$ in terms of the integer program (4.15)-(4.19):

$$\begin{aligned}
\text{minimize} \quad & q && && (4.15)\\
& n_1(s') = k' && && (4.16)\\
& \mathcal{H}(s', s'_i) \leq q - \mathcal{H}(s''_{\mathrm{ALG}}, s''_i) && \forall_{i\in\{1,2,\ldots,n\}} && (4.17)\\
& q \geq 0 && && (4.18)\\
& s'[j] \in \{0,1\} && \forall_{j\in\{1,2,\ldots,\beta\}} && (4.19)
\end{aligned}$$

where $Y \subseteq S, k = k' + k''$, and s''_{ALG} is the k''-completion of p''_Y. Recall that $\beta = p_Y^{(*)}$ and p''_Y is the "no-star part" of the pattern p_Y.

In the LP relaxation $\mathrm{LP}(Y, k')$ the constraint (4.19) is replaced with:

$$s'[j] \in [0,1] \qquad \forall_{j\in\{1,2,\ldots,\beta\}} \qquad (4.20)$$

Constraints (4.15)-(4.18),(4.20) are linear because

$$n_1(s') = \sum_{j=1}^{\beta} s'[j]$$

and

$$\mathcal{H}(s', s'_i) = \sum_{j=1}^{\beta} \Big(\chi(s'_i[j] = 0) \cdot s'[j] + \chi(s'_i[j] = 1) \cdot (1 - s'[j]) \Big)$$

are linear functions of $s'[j]$, where $j \in \{1, 2, \ldots, \beta\}$.

Lemma 38. *For each $R \in \mathbb{N}_{\geq 1}, Y \subseteq S, |Y| \leq R, k' \in \mathbb{N}, \epsilon_2 \in (0, \frac{1}{2})$ we can find an $(1 + 2\epsilon_2)$-approximation solution of $\mathrm{IP}(Y, k')$ by solving $\mathrm{LP}(Y, k')$ and considering at most*

$$(3n)^{\frac{3R\ln(2)}{(\epsilon_2)^2}} + m^{\frac{3R^2\ln(6)}{(\epsilon_2)^2}} \quad \textit{cases.}$$

Proof. Let us fix constant $\epsilon_2 \in (0, \frac{1}{2})$. We consider three cases:

Case 1: $\beta \leq \frac{3R\ln(3n)}{(\epsilon_2)^2}$
There is 2^β possibilities for s'. We can bound it as follows.

$$2^\beta \leq 2^{\frac{3R\ln(3n)}{(\epsilon_2)^2}} = e^{\ln(3n)\frac{3R\ln(2)}{(\epsilon_2)^2}} = (3n)^{\frac{3R\ln(2)}{(\epsilon_2)^2}} \in \mathrm{poly}(n),$$

because ϵ_2 and R are fixed constants. So we will check (in polynomial time) all possibilities for s' and we will find optimal solution of the integer program.

Case 2: $k' \leq \frac{3R^2 \ln(6)}{(\epsilon_2)^2}$

We can upper bound the number of possibilities for s' by the number of setting 1's into β positions:

$$\binom{\beta}{k'} \leq \beta^{k'} \leq \beta^{\frac{3R^2 \ln(6)}{(\epsilon_2)^2}} \leq m^{\frac{3R^2 \ln(6)}{(\epsilon_2)^2}} \in \text{poly}(m),$$

because ϵ_2 and R are fixed constants.

Case 3: $\beta > \frac{3R \ln(3n)}{(\epsilon_2)^2} \ \wedge \ k' > \frac{3R^2 \ln(6)}{(\epsilon_2)^2}$

We denote an optimal solution of the $\text{IP}(Y, k')$ by $((s')^{\text{IP}}, q^{\text{IP}})$ and an optimal solution of the $\text{LP}(Y, k')$ by $((s')^{\text{LP}}, q^{\text{LP}})$. Obviously we have $q^{\text{LP}} \leq q^{\text{IP}}$. We can solve the LP in polynomial time but we may obtain a fractional solution. We will round the variables independently. We will use a randomized rounding defined by distributions on each position $j \in \{1, 2, \ldots, \beta\}$:

$$\Pr\left[s'[j] = 1\right] = (s')^{\text{LP}}[j], \quad \Pr\left[s'[j] = 0\right] = 1 - (s')^{\text{LP}}[j]. \tag{4.21}$$

We can estimate the expected value of a distance to such a random solution s':

$$\forall_{i \in \{1,2,\ldots,n\}} \quad \mathbb{E}\Big[\mathcal{H}(s', s'_i)\Big] \overset{\text{def.}}{=} \mathbb{E}\left[\sum_{j=1}^{\beta} \Big|s'[j] - s'_i[j]\Big|\right]$$

$$= \mathbb{E}\left[\sum_{j=1}^{\beta}\Big(\quad \chi(s'_i[j] = 0) \cdot s'[j] \quad + \quad \chi(s'_i[j] = 1) \cdot (1 - s'[j]) \quad \Big)\right]$$

$$\overset{\text{lin. of } \mathbb{E}}{=} \sum_{j=1}^{\beta}\Big(\quad \chi(s'_i[j] = 0) \cdot \mathbb{E}\Big[s'[j]\Big] \quad + \quad \chi(s'_i[j] = 1) \cdot \mathbb{E}\Big[1 - s'[j]\Big] \quad \Big)$$

$$\overset{(4.21)}{=} \sum_{j=1}^{\beta}\Big(\quad \chi(s'_i[j] = 0) \cdot (s')^{\text{LP}}[j] \quad + \quad \chi(s'_i[j] = 1) \cdot \Big(1 - (s')^{\text{LP}}[j]\Big) \Big)$$

$$\overset{\text{def.}}{=} \mathcal{H}\Big((s')^{\text{LP}}, s'_i\Big) \overset{(4.17)}{\leq} q^{\text{LP}} - \mathcal{H}(s''_{\text{ALG}}, s''_i). \tag{4.22}$$

$\mathcal{H}(s', s'_i)$ is a sum of β independent binary random variables. For $\epsilon' \in (0, 1)$ using Chernoff-Hoeffding bounds we have:

$$\Pr\Big[\mathcal{H}(s', s'_i) \geq (1 + \epsilon') \cdot \mathbb{E}\Big[\mathcal{H}(s', s'_i)\Big]\Big] \overset{(4.1)}{\leq} \exp\left(-\frac{1}{3}(\epsilon')^2 \cdot \mathbb{E}\Big[\mathcal{H}(s', s'_i)\Big]\right).$$

If we take $\epsilon' = \frac{\epsilon_2 \cdot q^{\text{IP}}}{\mathbb{E}[\mathcal{H}(s', s'_i)]}$ then we obtain:

$$\exp\left(-\frac{1}{3} \cdot \frac{(\epsilon_2)^2 \cdot (q^{\text{IP}})^2}{\mathbb{E}\Big[\mathcal{H}(s', s'_i)\Big]}\right) \geq \Pr\Big[\mathcal{H}(s', s'_i) \geq \mathbb{E}\Big[\mathcal{H}(s', s'_i)\Big] + \epsilon_2 \cdot q^{\text{IP}}\Big]$$

$$\overset{(4.22)}{\geq} \Pr\left[\mathcal{H}(s', s'_i) \geq q^{\mathrm{LP}} - \mathcal{H}(s''_{\mathrm{ALG}}, s''_i) + \epsilon_2 \cdot q^{\mathrm{IP}}\right]. \tag{4.23}$$

We want to know an upper bound for the probability that we make an error greater than $\epsilon_2 \cdot q^{\mathrm{IP}}$ for at least one vote:

$$\begin{aligned}&\Pr\left[\exists_{i\in\{1,2,\ldots,n\}} : \mathcal{H}(s', s'_i) \geq q^{\mathrm{LP}} - \mathcal{H}(s''_{\mathrm{ALG}}, s''_i) + \epsilon_2 \cdot q^{\mathrm{IP}}\right] \\ &\overset{(4.23)}{\leq} n \cdot \exp\left(-\frac{1}{3} \cdot \frac{(\epsilon_2)^2 \cdot (q^{\mathrm{IP}})^2}{\mathbb{E}\left[\mathcal{H}(s', s'_i)\right]}\right) \leq n \cdot \exp\left(-\frac{1}{3}(\epsilon_2)^2 \cdot q^{\mathrm{IP}}\right),\end{aligned} \tag{4.24}$$

where the last inequality is because of:

$$\mathbb{E}\left[\mathcal{H}(s', s'_i)\right] \overset{(4.22)}{\leq} q^{\mathrm{LP}} - \mathcal{H}(s''_{\mathrm{ALG}}, s''_i) \leq q^{\mathrm{IP}}.$$

We want to further upper bound the probability in (4.24). From the assumption about β and from Lemma 36 we have:

$$\frac{3R\ln(3n)}{(\epsilon_2)^2} < \beta \overset{\text{Lemma 36}}{\leq} |Y| \cdot \mathrm{OPT} \leq R \cdot \mathrm{OPT} \leq R \cdot q^{\mathrm{IP}},$$

equivalently

$$\frac{1}{3} > n \cdot \exp\left(-\frac{1}{3}(\epsilon_2)^2 \cdot q^{\mathrm{IP}}\right). \tag{4.25}$$

Finally we have

$$\Pr\left[\exists_{i\in\{1,2,\ldots,n\}} : \mathcal{H}(s', s'_i) \geq q^{\mathrm{LP}} - \mathcal{H}(s''_{\mathrm{ALG}}, s''_i) + \epsilon_2 \cdot q^{\mathrm{IP}}\right] \overset{(4.24),(4.25)}{<} \frac{1}{3}. \tag{4.26}$$

Hence, with probability at least $\frac{2}{3}$ we obtain:

$$\begin{aligned}\forall_{i\in\{1,2,\ldots,n\}} \quad &\mathcal{H}(s' \cdot s''_{\mathrm{ALG}}, s_i) = \mathcal{H}(s', s'_i) + \mathcal{H}(s''_{\mathrm{ALG}}, s''_i) \\ &\overset{(4.26)}{<} q^{\mathrm{LP}} - \mathcal{H}(s''_{\mathrm{ALG}}, s''_i) + \epsilon_2 \cdot q^{\mathrm{IP}} + \mathcal{H}(s''_{\mathrm{ALG}}, s''_i) \leq (1+\epsilon_2) \cdot q^{\mathrm{IP}}.\end{aligned} \tag{4.27}$$

We can also obtain a wrong number o 1's in s'. We define s'_{ALG} as the k'-completion of s'. We will show that the additional error for such operation is not so big. Expected number of 1's in s' is equal k':

$$\mathbb{E}\left[n_1(s')\right] \overset{\text{def.}}{=} \mathbb{E}\left[\sum_{j=1}^{\beta} s'[j]\right] \overset{\text{lin. of } \mathbb{E}}{=} \sum_{j=1}^{\beta} (s')^{\mathrm{LP}}[j] \overset{\text{def.}}{=} n_1\left((s')^{\mathrm{LP}}\right) \overset{(4.16)}{=} k'.$$

We want to know how much we lose taking the k'-completion. Similar as before, $n_1(s') = \sum_{j=1}^{\beta} s'[j]$ is a sum of β independent binary random variables. For $\epsilon'' \in (0,1)$ using Chernoff-Hoeffding bounds we have:

$$\Pr\left[n_1(s') \geq (1+\epsilon'') \cdot k'\right] \overset{(4.1)}{\leq} \exp\left(-\frac{1}{3}(\epsilon'')^2 \cdot k'\right),$$

$$\Pr\left[n_1(s') \leq (1-\epsilon'')\cdot k'\right] \overset{(4.2)}{\leq} \exp\left(-\frac{1}{2}(\epsilon'')^2\cdot k'\right).$$

Taking both inequalities together, $\epsilon'' = \frac{\epsilon_2}{R}$ and using assumption $k' > \frac{3R^2\ln(6)}{(\epsilon_2)^2}$ we get:

$$\Pr\left[|n_1(s') - k'| \geq \epsilon''\cdot k'\right] \leq 2\cdot\exp\left(-\frac{1}{3}(\epsilon'')^2\cdot k'\right) \leq 2\cdot\exp\left(-\frac{1}{3}\frac{(\epsilon_2)^2}{R^2}\cdot k'\right) < \frac{1}{3}.$$

So with probability at least $\frac{2}{3}$ the error from taking the k'-completion is not greater than $\epsilon''\cdot k' = \frac{\epsilon_2}{R}\cdot k' \leq \frac{\epsilon_2}{R}\cdot\beta \overset{\text{Lemma 36}}{\leq} \frac{\epsilon_2}{R}\cdot|Y|\cdot\mathrm{OPT} \leq \epsilon_2\cdot\mathrm{OPT} \leq \epsilon_2\cdot q^{\mathrm{IP}}$.

Combining the above with (4.27) we obtain an $(1+2\epsilon_2)$-approximate solution with probability at least $\frac{1}{3}$. □

4.1.4 Algorithm and Its Complexity Analysis

Now we are ready to combine the ideas into a single algorithm (see Algorithm 4).

Algorithm 4: Polynomial time approximation scheme for Minimax Approval Voting.

1 **for** *each* R*-element subset* $Y = \{s_{i_1}, s_{i_2}, \ldots, s_{i_R}\} \subseteq S$ **do**
2 **for** *each division* k *into two parts* $k = k' + k''$ **do**
3 $s''_{\mathrm{ALG}} \leftarrow k''$-completion of p''_Y (if not possible, then skip this inner iteration);
4 $s'_{\mathrm{ALG}} \leftarrow$ an approximation solution of $\mathrm{IP}(Y, k')$ from Lemma 38 (if $\mathrm{LP}(Y, k')$ infeasible, then skip this inner iteration);
5 evaluate $s'_{\mathrm{ALG}}\cdot s''_{\mathrm{ALG}}$ by computing $\max_{i\in\{1,2,\ldots,n\}}\mathcal{H}(s_i, s'_{\mathrm{ALG}}\cdot s''_{\mathrm{ALG}})$;
6 $s_{\mathrm{ALG}} \leftarrow$ the best solution $s'_{\mathrm{ALG}}\cdot s''_{\mathrm{ALG}}$ from a loop in lines 1-5;
7 **return** s_{ALG};

It remains to argue that for a large enough parameter R the above algorithm will at some point consider a stable subset of votes X that leads to an accurate enough approximation of the objective function of Minimax Approval Voting.

Theorem 39. *For any $\epsilon > 0$ we can find $(1+\epsilon)$-approximation solution for* Minimax Approval Voting *in polynomial time $n^{\mathcal{O}(1/\epsilon^4)}\cdot m^{\mathcal{O}(1)} + n^{\mathcal{O}(1/\epsilon)}\cdot m^{\mathcal{O}(1/\epsilon^4)}$ with probability at least $1-p$, for any fixed $p > 0$.*

Proof. Let $\epsilon_0 = \frac{\epsilon}{3} < \frac{1}{3}$.

By Lemma 35, there exists an $\frac{\epsilon_0\cdot\mathrm{OPT}}{2}$-stable set of votes $X \subseteq S$ of cardinality $|X| = R = \lceil\frac{2}{\epsilon_0}\rceil$.

Consider algorithm Algorithm 4. In one iteration it will consider X and k', k'' such that $n_1(s'_{\mathrm{OPT}}) = k'$. Recall that s''_{ALG} is the specific k''-completion of p''_X. By Lemma 37 we have:

$$\mathcal{H}(s'_{\mathrm{OPT}}\cdot s''_{\mathrm{ALG}}, s_i) \leq (1+\epsilon_0)\cdot\mathrm{OPT},$$

hence $\left(s' = s'_{\mathrm{OPT}}, q = (1+\epsilon_0)\cdot \mathrm{OPT}\right)$ is a feasible solution to $\mathrm{IP}(X,k')$ and the optimal value of $\mathrm{IP}(X,k')$ is at most $(1+\epsilon_0)\cdot \mathrm{OPT}$.

By Lemma 38 with $\epsilon_2 = \frac{\epsilon_0}{2}$ we find a $(1+\epsilon_0)$-approximate solution $(s'_{\mathrm{ALG}}, q_{\mathrm{ALG}})$ to $\mathrm{IP}(X,k')$ with probability at least $\frac{1}{3}$. So we have:

$$q_{\mathrm{ALG}} \leq (1+\epsilon_0)\cdot(1+\epsilon_0)\cdot \mathrm{OPT} \overset{\epsilon_0<1}{\leq} (1+3\epsilon_0)\cdot \mathrm{OPT} = (1+\epsilon)\cdot \mathrm{OPT}.$$

It remains to observe, that $s_{\mathrm{ALG}} = s'_{\mathrm{ALG}} \cdot s''_{\mathrm{ALG}}$ is a solution to Minimax Approval Voting of cost $q_{\mathrm{ALG}} \leq (1+\epsilon)\cdot \mathrm{OPT}$ with probability at least $\frac{1}{3}$. In order to increase the success probablility to $1-p$ we repeat the algorithm $\log_{2/3}(p) = \mathcal{O}(1)$ times. Then indeed, probability of incorrect answer is at most $(\frac{2}{3})^{\log_{2/3}(p)} = p$.

The algorithm examined $\mathcal{O}(n^R) \subseteq \mathcal{O}\left(n^{\lceil \frac{6}{\epsilon} \rceil}\right) \subseteq \mathrm{poly}(n)$ subsets Y, $\mathcal{O}(m)$ choices of k' and each time considered at most $\mathcal{O}\left((3n)^{108\cdot\lceil 6/\epsilon\rceil\cdot\ln(2)/\epsilon^2} + m^{108\cdot\lceil 6/\epsilon\rceil^2\cdot\ln(6)/\epsilon^2}\right)$ cases. So the total running time is upper bounded by $n^{\mathcal{O}(1/\epsilon^4)}\cdot m^{\mathcal{O}(1)} + n^{\mathcal{O}(1/\epsilon)}\cdot m^{\mathcal{O}(1/\epsilon^4)}$. □

4.2 Parameterized Approximation Scheme

In this section we will show a parameterized time approximation scheme for Minimax Approval Voting proving the following theorem.

Theorem 40. *There exists a randomized algorithm which, given an instance $(S = \{s_i\}_{i\in[n]}, k, d)$ of the decision version of* Minimax Approval Voting *(d is the required maximal distance) and any $\epsilon \in (0,3)$, runs in time $\mathcal{O}\left(\left(\frac{3}{\epsilon}\right)^{2d}\cdot(m+n) + mn\right)$ and either*

(i) reports a solution at a distance at most $(1+\epsilon)d$ from S, or

(ii) reports that there is no solution at a distance at most d from S.

In the latter case, the answer is correct with probability at least $1-p$, for arbitrarily small fixed $p > 0$.

Let us proceed with the proof. In what follows we assume $p = 1/2$, since then we can get the claim even if $p < 1/2$ by repeating the whole algorithm $\lceil \log_2(1/p) \rceil$ times. Indeed, then the algorithm returns an incorrect answer only if each of the $\lceil \log_2(1/p) \rceil$ repetitions returned an incorrect answer, which happens with probability at most $(1/2)^{\log_2(1/p)} = p$.

Assume we are given a yes-instance and let us fix a solution $s^* \in S_{k,m}$, i.e., a string at distance at most d from all the input strings. Our approach is to begin with a string $x_0 \in S_{k,m}$ not very far from s^*, and next perform a number of steps. In the j-th step we either conclude that x_{j-1} is already a $(1+\epsilon)$-approximate solution, or with some probability we find another string x_j which is closer to s^*.

First observe that if $|n_1(s_1) - k| > d$, then clearly there is no solution and our algorithm reports NO. Hence in what follows we assume $|n_1(s_1) - k| \leq d$. We set x_0

to be any k-completion of s_1, therefore we get $\mathcal{H}(x_0, s_1) \leq d$. Since $\mathcal{H}(s_1, s^*) \leq d$, by the triangle inequality we get the following bound

$$\mathcal{H}(x_0, s^*) \leq \mathcal{H}(x_0, s_1) + \mathcal{H}(s_1, s^*) \leq 2d. \tag{4.28}$$

Now we are ready to describe our algorithm precisely (see also Algorithm 5). We begin with x_0 defined as above. We are going to create a sequence of strings $x_0, x_1, \ldots, x_d$ satisfying $n_1(x_j) = k$ for every j. For $j \in [d]$ we do the following. If for every $i \in [n]$ we have $\mathcal{H}(x_{j-1}, s_i) \leq (1+\epsilon)d$ the algorithm terminates and returns x_{j-1}. Otherwise, fix any $i \in [n]$ such that $\mathcal{H}(x_{j-1}, s_i) > (1+\epsilon)d$. Let $P_{j,0} = \{a \in [m] : 0 = x_{j-1}[a] \neq s_i[a] = 1\}$ and $P_{j,1} = \{a \in [m] : 1 = x_{j-1}[a] \neq s_i[a] = 0\}$. The algorithm samples a position $a_0 \in P_{j,0}$ and a position $a_1 \in P_{j,1}$. In case $P_{j,0} = \emptyset$ or $P_{j,1} = \emptyset$ we return NO because it means that $\mathcal{H}(s_i, S_{k,m}) = \mathcal{H}(s_i, x_{j-1}) > d$. Then, x_j is obtained from x_{j-1} by swapping the 0 at position a_0 with the 1 at position a_1. If the algorithm finishes without finding a solution, it reports NO.

Algorithm 5: Parameterized approximation scheme for Minimax Approval Voting.

```
if |n_1(s_1) − k| > d then return NO;
x_0 ← any k-completion of s_1;
for j ∈ {1, 2, ..., d} do
    if H(x_{j−1}, S) ≤ (1 + ε)d then return x_{j−1};
    otherwise there exists s_i s.t. H(x_{j−1}, s_i) > (1 + ε)d;
    P_{j,0} ← {a ∈ [m] : 0 = x_{j−1}[a] ≠ s_i[a] = 1};
    P_{j,1} ← {a ∈ [m] : 1 = x_{j−1}[a] ≠ s_i[a] = 0};
    if min(|P_{j,0}|, |P_{j,1}|) = 0 then return NO;
    x_j ← swap 0 and 1 in x_{j−1} on a pair of random positions from P_{j,0} and P_{j,1};
if H(x_d, S) ≤ (1 + ε)d then return x_d;
else return NO;
```

The following lemma is the key to get a lower bound on the probability that the x_j's get close to s^*.

Lemma 41. *Let x be a string in $S_{k,m}$ such that $\mathcal{H}(x, s_i) \geq (1+\epsilon)d$ for some $i \in [n]$. Let $s^* \in S_{k,m}$ be any solution, i.e., a string at distance at most d from all the strings s_j, $j \in [n]$. Denote*

$$P_0^* = \{a \in [m] : 0 = x[a] \neq s_i[a] = s^*[a] = 1\},$$

$$P_1^* = \{a \in [m] : 1 = x[a] \neq s_i[a] = s^*[a] = 0\}.$$

Then, it holds that $\min(|P_0^*|, |P_1^*|) \geq \frac{\epsilon d}{2}$.

Proof. Let P be the set of positions on which x and s_i differ, i.e., $P = \{a \in [m] : x[a] \neq s_i[a]\}$ (see Figure 4.2). Note that $P_0^* \cup P_1^* \subseteq P$. Let $Q = [m] \setminus P$.

The intuition behind the proof is that if $\min(|P_0^*|, |P_1^*|)$ is small, then s^* differs too much from s_i, either because $s^*|_P$ is similar to $x|_P$ (when $|P_0^*| \approx |P_1^*|$) or because $s^*|_Q$ has much more 1's than $s_i|_Q$ (when $|P_0^*|$ differs much from $|P_1^*|$).

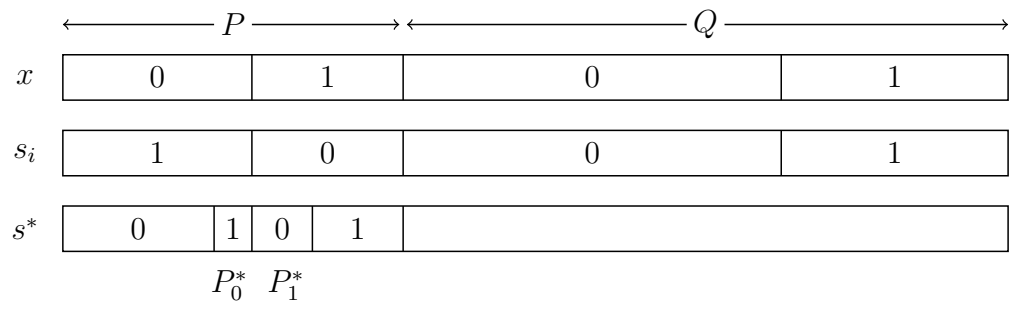


Figure 4.2: Strings x, s_i and s^* after permuting the positions.

We begin with a couple of useful observations on the number of 1's in different parts of x, s_i and s^*. Since x and s_i are the same on Q, we get

$$n_1(x|_Q) = n_1(s_i|_Q). \tag{4.29}$$

Since $n_1(x) = n_1(s^*)$, we get $n_1(x|_P) + n_1(x|_Q) = n_1(s^*|_P) + n_1(s^*|_Q)$, and further

$$n_1(s^*|_Q) - n_1(x|_Q) = n_1(x|_P) - n_1(s^*|_P). \tag{4.30}$$

Finally note that

$$n_1(s^*|_P) = |P_0^*| + n_1(x|_P) - |P_1^*|. \tag{4.31}$$

We are going to derive a lower bound on $\mathcal{H}(s_i, s^*)$. First, we have

$$\begin{aligned}\mathcal{H}(s_i|_P, s^*|_P) = |P| - (|P_0^*| + |P_1^*|) = \mathcal{H}(x, s_i) - (|P_0^*| + |P_1^*|)\\ \geq (1+\epsilon)d - (|P_0^*| + |P_1^*|).\end{aligned} \tag{4.32}$$

On the other hand, it holds that

$$\begin{aligned}\mathcal{H}(s_i|_Q, s^*|_Q) \geq |n_1(s^*|_Q) - n_1(s_i|_Q)| \overset{(4.29)}{=} |n_1(s^*|_Q) - n_1(x|_Q)|\\ \overset{(4.30)}{=} |n_1(x|_P) - n_1(s^*|_P)| \overset{(4.31)}{=} \Big||P_1^*| - |P_0^*|\Big|.\end{aligned} \tag{4.33}$$

It follows that

$$\begin{aligned}d \quad &\geq \quad \mathcal{H}(s_i, s^*) = \mathcal{H}(s_i|_P, s^*|_P) + \mathcal{H}(s_i|_Q, s^*|_Q)\\ &\overset{(4.32),(4.33)}{\geq} (1+\epsilon)d - (|P_0^*| + |P_1^*|) + ||P_1^*| - |P_0^*|| = (1+\epsilon)d - 2\min(|P_0^*|, |P_1^*|).\end{aligned}$$

Hence, $\min(|P_0^*|, |P_1^*|) \geq \frac{\epsilon d}{2}$ as required. □

Corollary 42. *Assume that there is a solution $s^* \in S_{k,m}$ and that the algorithm created a string x_j, for some $j \in \{0, \ldots, d\}$. Then, it holds that* $\Pr[\mathcal{H}(x_j, s^*) \leq 2d - 2j] \geq \left(\frac{\epsilon}{3}\right)^{2j}$.

Proof. We use induction on j. For $j = 0$ the claim follows from (4.28). Consider $j > 0$. By the induction hypothesis, we get

$$\Pr[\mathcal{H}(x_{j-1}, s^*) \leq 2d - 2j + 2] \geq \left(\frac{\epsilon}{3}\right)^{2j-2}. \tag{4.34}$$

Assume that $\mathcal{H}(x_{j-1}, s^*) \leq 2d - 2j + 2$. Since x_j was created, $\mathcal{H}(x_{j-1}, s_i) > (1 + \epsilon)d$ for some $i \in [n]$. Since $\mathcal{H}(s^*, s_i) \leq d$, by the triangle inequality we get the following

$$|P_{j,0}| + |P_{j,1}| = \mathcal{H}(x_{j-1}, s_i) \leq \mathcal{H}(x_{j-1}, s^*) + \mathcal{H}(s^*, s_i) \leq 3d - 2j + 2 \leq 3d. \tag{4.35}$$

Then, we have

$$\Pr[\mathcal{H}(x_j, s^*) \leq 2d - 2j \mid \mathcal{H}(x_{j-1}, s^*) \leq 2d - 2j + 2] \geq \frac{|P_0^*| \cdot |P_1^*|}{|P_{j,0}| \cdot |P_{j,1}|} \geq \frac{\left(\frac{\epsilon d}{2}\right)^2}{\left(\frac{3d}{2}\right)^2} = \left(\frac{\epsilon}{3}\right)^2. \tag{4.36}$$

The first inequality holds through counting proper swaps among all possible swaps. The second inequality follows from Lemma 41 and (4.35). The claim follows by combining (4.34) and (4.36). □

In order to increase the success probability, we repeat the algorithm until a solution is found or the number of repetitions is at least $(3/\epsilon)^{2d}$. By Corollary 42 the probability that there is a solution but it was not found is bounded by

$$\left(1 - \left(\frac{\epsilon}{3}\right)^{2d}\right)^{(3/\epsilon)^{2d}} = \left(1 - \frac{1}{(3/\epsilon)^{2d}}\right)^{(3/\epsilon)^{2d}} \leq \frac{1}{e} < \frac{1}{2}.$$

This finishes the proof of Theorem 40.

p \ d	10	15	20	25
0.5	0.3	0.2	0.15	0.12
10^{-10}	0.357	0.225	0.164	0.129
10^{-20}	0.370	0.230	0.167	0.131

Table 4.2: Rounded values of $\epsilon = \frac{3}{d} \cdot \left(\log \frac{1}{p}\right)^{\frac{1}{2d}}$.

Table 4.2 presents (rounded) values of ϵ for which the worst case bounds (with constants omitted) for the running times of algorithm from Theorem 40 and the algorithm of [116] are equal, i.e., when $(3/\epsilon)^{2d} \cdot \log_2(1/p) = d^{2d}$ which gives $\epsilon = (3/d) \cdot (\log_2(1/p))^{\frac{1}{2d}}$. For ϵ greater than the values in Table 4.2 our algorithm can be faster than the previous one for instances with no solution at distance at most d from S. Note that the effect of p on the border value of ϵ is not very significant. However, a meaningful comparison of practical aspects of these two algorithms requires performing a series of experiments with actual implementations.

4.3 A Faster Polynomial Time Approximation Scheme

The goal of this section is to present a PTAS for the optimization version of Minimax Approval Voting running in time $n^{\mathcal{O}(1/\epsilon^2 \cdot \log(1/\epsilon))} \cdot \text{poly}(m)$. It is achieved by combining the parameterized approximation scheme from Theorem 40 with the following result, which might be of independent interest.

Theorem 43. *There exists a randomized polynomial time algorithm which, for arbitrarily small fixed $p > 0$, given an instance $(\{s_i\}_{i\in[n]}, k)$ of* Minimax Approval Voting *and any $\epsilon > 0$ such that $\text{OPT} \geq \frac{122 \ln n}{\epsilon^2}$, reports a solution, which with probability at least $1 - p$ is at distance at most $(1+\epsilon) \cdot \text{OPT}$ from S.*

In what follows, we prove Theorem 43. As in the proof of Theorem 40 we assume w.l.o.g. $p = 1/2$. Note that we can assume $\epsilon < 1$, for otherwise it suffices to use the 2-approximation of [39]. We also assume $n \geq 3$, for otherwise it is a straightforward exercise to find an optimal solution in linear time. Let us define a linear program (4.37–4.40):

$$\text{minimize} \quad d \tag{4.37}$$

$$\sum_{j\in[m]} x_j = k \tag{4.38}$$

$$\sum_{\substack{j\in[m]\\ s_i[j]=1}} (1 - x_j) + \sum_{\substack{j\in[m]\\ s_i[j]=0}} x_j \leq d \qquad \forall i \in [n] \tag{4.39}$$

$$x_j \in [0,1] \qquad \forall j \in [m] \tag{4.40}$$

The linear program (4.37–4.40) is a relaxation of the natural integer program for Minimax Approval Voting, obtained by replacing (4.40) by the discrete constraint $x_j \in \{0,1\}$. Indeed, observe that x_j corresponds to the j-th letter of the solution $x = x_1 \cdots x_m$, (4.38) states that $n_1(x) = k$, and (4.39) states that $\mathcal{H}(x, S) \leq d$.

Algorithm 6: The algorithm from Theorem 43

```
Solve the LP (4.37–4.40) obtaining an optimal solution (x*_1, ..., x*_m, d*);
for j ∈ {1, 2, ..., m} do
    Set x[j] ← 1 with probability x*_j and x[j] ← 0 with probability 1 − x*_j;
y ← any k-completion of x;
return y;
```

Our algorithm is defined as follows (see Algorithm 6). First we solve the linear program in time $\text{poly}(n, m)$ using the interior point method [92]. Let $(x_1^*, \ldots, x_m^*, d^*)$ be the obtained optimal solution. Clearly, $d^* \leq \text{OPT}$. We randomly construct a string $x \in \{0,1\}^m$, guided by the values x_j^*. More precisely, for every $j \in [m]$ independently, we set $x[j] = 1$ with probability x_j^*. Note that x does not need to contain k ones. Let y be any k-completion of x. The algorithm returns y.

Clearly, the above algorithm runs in polynomial time. In what follows we bound the probability of error. To this end we prove upper bounds on the probability that x is far from S and the probability that the number of 1's in x is far from k. This is done in Lemmas 44 and 45, which can be shown using standard Chernoff-Hoeffding bounds.

Lemma 44. *It holds that* $\Pr\left[\mathcal{H}(x,S) > (1+\frac{\epsilon}{2})\cdot \mathrm{OPT}\right] \leq \frac{1}{4}$.

Proof. For every $i \in [n]$ we define a random variable D_i that measures the distance between x^* and s_i by

$$D_i = \sum_{\substack{j\in[m]\\ s_i[j]=1}} (1-x[j]) + \sum_{\substack{j\in[m]\\ s_i[j]=0}} x[j].$$

Note that $x[i]$ are independent binary random variables. Using linearity of the expectation we obtain

$$\begin{aligned}
\mathbb{E}[D_i] = \quad & \mathbb{E}\left[\sum_{j\in[m],s_i[j]=1} (1-x[j]) \quad + \sum_{j\in[m],s_i[j]=0} x[j] \right] \\
= \quad & \sum_{j\in[m],s_i[j]=1} (1-\mathbb{E}[x[j]]) \quad + \sum_{j\in[m],s_i[j]=0} \mathbb{E}[x[j]] \\
= \quad & \sum_{j\in[m],s_i[j]=1} (1-x^*_j) \quad + \sum_{j\in[m],s_i[j]=0} x^*_j \qquad \leq d^* \leq \mathrm{OPT}. \qquad (4.41)
\end{aligned}$$

Note that D_i is a sum of m independent binary random variables $X_j = 1-x[j]$ when $s_i[j]=1$ and $X_j = x[j]$ otherwise. Denote $\delta = \epsilon \cdot \frac{\mathrm{OPT}}{2\mathbb{E}[D_i]}$. We apply Chernoff-Hoeffding bounds. For $\delta < 1$ we have

$$\begin{aligned}
\Pr[D_i > \left(1+\tfrac{\epsilon}{2}\right)\cdot \mathrm{OPT}] &\overset{(4.41)}{\leq} \Pr\left[D_i > \mathbb{E}[D_i] + \tfrac{\epsilon}{2}\cdot \mathrm{OPT}\right] = \Pr\left[D_i > (1+\delta)\cdot \mathbb{E}[D_i]\right] \\
&\overset{(4.1)}{\leq} \exp\left(-\frac{1}{3}\left(\epsilon\cdot\frac{\mathrm{OPT}}{2\mathbb{E}[D_i]}\right)^2 \mathbb{E}[D_i]\right) \overset{(4.41)}{\leq} \exp\left(-\frac{\epsilon^2\cdot\mathrm{OPT}}{12}\right).
\end{aligned}$$

In case $\delta \geq 1$ we proceed analogously, using Chernoff-Hoeffding bounds (4.3) we get

$$\Pr[D_i > \left(1+\tfrac{\epsilon}{2}\right)\cdot \mathrm{OPT}] \overset{(4.3)}{\leq} \exp\left(-\frac{\epsilon\cdot\mathrm{OPT}}{6}\right) \overset{1>\epsilon}{\leq} \exp\left(-\frac{\epsilon^2\cdot\mathrm{OPT}}{12}\right).$$

Next, we use the union bound to get the claim

$$\begin{aligned}
&\Pr\left[\mathcal{H}(x,S) > (1+\tfrac{\epsilon}{2})\cdot\mathrm{OPT}\right] = \Pr\left[\exists_{i\in[n]} \quad D_i > \left(1+\tfrac{\epsilon}{2}\right)\cdot\mathrm{OPT}\right] \\
&\leq n\cdot\exp\left(-\frac{\epsilon^2\cdot\mathrm{OPT}}{12}\right) \leq n\cdot\exp\left(-\frac{\frac{122\ln n}{\mathrm{OPT}}\cdot\mathrm{OPT}}{12}\right) < n^{-9} \overset{n\geq 3}{<} \frac{1}{4}.
\end{aligned}$$

□

Lemma 45. *It holds that* $\Pr\left[|n_1(x)-k| > \frac{\epsilon}{2}\cdot \mathrm{OPT}\right] \leq \frac{1}{4}$.

Proof. First we note that

$$\mathbb{E}[n_1(x)] = \mathbb{E}\Big[\sum_{j\in[m]} x[j]\Big] = \sum_{j\in[m]} \mathbb{E}[x[j]] = \sum_{j\in[m]} x^*_j = k. \tag{4.42}$$

Pick an $i \in [n]$. Define the random variables

$$E_i = \sum_{j\in[m], s_i[j]=1} (1-x[j]), \quad F_i = \sum_{j\in[m], s_i[j]=0} x[j].$$

Let $D_i = E_i + F_i$, as in the proof of Lemma 44. By (4.41) we have

$$\mathbb{E}[E_i] \leq \mathbb{E}[E_i] + \mathbb{E}[F_i] = \mathbb{E}[D_i] \leq \mathrm{OPT} \tag{4.43}$$

$$\mathbb{E}[F_i] \leq \mathbb{E}[E_i] + \mathbb{E}[F_i] = \mathbb{E}[D_i] \leq \mathrm{OPT} \tag{4.44}$$

Both E_i and F_i are sums of independent binary random variables and we apply Chernoff-Hoeffding bounds as follows. When $\frac{1}{4}\epsilon\cdot\frac{\mathrm{OPT}}{\mathbb{E}[E_i]} \leq 1$ then using (4.1) and (4.2) we obtain

$$\begin{aligned}
&\Pr\left[\left|E_i - \mathbb{E}[E_i]\right| > \frac{1}{4}\epsilon\cdot\mathrm{OPT}\right] \\
&\overset{(4.1),(4.2)}{\leq} \exp\left(-\frac{1}{3}\cdot\frac{1}{16}\epsilon^2\cdot\frac{(\mathrm{OPT})^2}{\mathbb{E}^2[E_i]}\cdot\mathbb{E}[E_i]\right) + \exp\left(-\frac{1}{2}\cdot\frac{1}{16}\epsilon^2\cdot\frac{(\mathrm{OPT})^2}{\mathbb{E}^2[E_i]}\cdot\mathbb{E}[E_i]\right) \\
&\overset{(4.43)}{\leq} 2\cdot\exp\left(-\frac{1}{48}\epsilon^2\cdot\mathrm{OPT}\right),
\end{aligned}$$

otherwise $\left(\frac{1}{4}\epsilon\cdot\frac{\mathrm{OPT}}{\mathbb{E}[E_i]} > 1\right)$, using (4.3) and (4.4), we have

$$\begin{aligned}
&\Pr\left[\left|E_i - \mathbb{E}[E_i]\right| > \frac{1}{4}\epsilon\cdot\mathrm{OPT}\right] \overset{(4.3),(4.4)}{\leq} \exp\left(-\frac{1}{3}\cdot\frac{1}{4}\epsilon\cdot\frac{\mathrm{OPT}}{\mathbb{E}[E_i]}\cdot\mathbb{E}[E_i]\right) + 0 \\
&\leq \exp\left(-\frac{1}{12}\epsilon\cdot\mathrm{OPT}\right) \overset{1>\epsilon}{\leq} 2\cdot\exp\left(-\frac{1}{48}\epsilon^2\cdot\mathrm{OPT}\right).
\end{aligned}$$

To sum up, in both cases we have shown that

$$\Pr\left[\left|E_i - \mathbb{E}[E_i]\right| > \frac{\epsilon}{4}\cdot\mathrm{OPT}\right] \leq 2\cdot\exp\left(-\frac{1}{48}\epsilon^2\cdot\mathrm{OPT}\right). \tag{4.45}$$

Similarly we show

$$\Pr\left[\left|F_i - \mathbb{E}[F_i]\right| > \frac{\epsilon}{4}\cdot\mathrm{OPT}\right] \overset{(4.1),(4.2),(4.3),(4.4),(4.44)}{\leq} 2\cdot\exp\left(-\frac{1}{48}\epsilon^2\cdot\mathrm{OPT}\right). \tag{4.46}$$

We see that

$$n_1(x) = \sum_{j\in[m]} x[j] = n_1(s_i) - \sum_{j\in[m], s_i[j]=1}(1-x[j]) + \sum_{j\in[m], s_i[j]=0} x[j] = n_1(s_i) - E_i + F_i \tag{4.47}$$

and hence it holds

$$\mathbb{E}[n_1(x)] = n_1(s_i) - \mathbb{E}[E_i] + \mathbb{E}[F_i]. \tag{4.48}$$

Additionally we will use

$$\forall x, y \in \mathbb{R} \quad |x - y| > a \implies |x| > a/2 \ \vee \ |y| > a/2. \tag{4.49}$$

Now we can write

$$\begin{aligned}
&\Pr\left[\left|n_1(x) - k\right| > \tfrac{1}{2}\epsilon \cdot \mathrm{OPT}\right] \overset{(4.42)}{=} \Pr\left[\left|n_1(x) - \mathbb{E}[n_1(x)]\right| > \tfrac{1}{2}\epsilon \cdot \mathrm{OPT}\right] \\
&\overset{(4.47),(4.48)}{=} \Pr\left[\left|n_1(s_i) - E_i + F_i - n_1(s_i) + \mathbb{E}[E_i] - \mathbb{E}[F_i]\right| > \tfrac{1}{2}\epsilon \cdot \mathrm{OPT}\right] \\
&\overset{(4.49)}{\leq} \Pr\left[\left|E_i - \mathbb{E}[E_i]\right| > \tfrac{1}{4}\epsilon \cdot \mathrm{OPT} \quad \vee \quad \left|F_i - \mathbb{E}[F_i]\right| > \tfrac{1}{4}\epsilon \cdot \mathrm{OPT}\right] \\
&\leq \Pr\left[\left|E_i - \mathbb{E}[E_i]\right| > \tfrac{1}{4}\epsilon \cdot \mathrm{OPT}\right] + \Pr\left[\left|F_i - \mathbb{E}[F_i]\right| > \tfrac{1}{4}\epsilon \cdot \mathrm{OPT}\right] \\
&\overset{(4.45),(4.46)}{\leq} 4 \cdot \exp\left(-\tfrac{1}{48}\epsilon^2 \cdot \mathrm{OPT}\right) \overset{\text{assum.}}{\leq} 4 \cdot \exp\left(-\tfrac{122}{48}\ln n\right) \overset{n\geq 3}{<} \tfrac{1}{4}.
\end{aligned}$$

□

We can finish the proof of Theorem 43. By Lemmas 44 and 45 with probability at least $1/2$ both $\mathcal{H}(x, S) \leq (1 + \frac{1}{2}\epsilon) \cdot \mathrm{OPT}$ and $\mathcal{H}(y, x) = |n_1(x) - k| \leq \frac{1}{2}\epsilon \cdot \mathrm{OPT}$. By the triangle inequality this implies that $\mathcal{H}(y, S) \leq (1 + \epsilon) \cdot \mathrm{OPT}$, with probability at least $1/2$ as required.

We conclude the section by combining Theorems 40 and 43 to get a faster PTAS.

Theorem 46. *For any $\epsilon > 0$ we can find $(1+\epsilon)$-approximation solution for* Minimax Approval Voting *in polynomial time $n^{\mathcal{O}(1/\epsilon^2 \cdot \log(1/\epsilon))} \cdot m^{\mathcal{O}(1)}$ with probability at least $1 - r$, for any fixed $r > 0$.*

Proof. First we run the algorithm from Theorem 40 for $d = \lceil \frac{122 \ln n}{\epsilon^2} \rceil$ and $p = r/2$.

If it reports a solution, for every $d' \leq d$ we apply Theorem 40 with $p = r/2$ and we return the best solution. If $\mathrm{OPT} \geq d$, even the initial solution is at distance at most $(1 + \epsilon)d \leq (1 + \epsilon) \cdot \mathrm{OPT}$ from S. Otherwise, at some point $d' = \mathrm{OPT}$ and we get a $(1 + \epsilon)$-approximation with probability at least $1 - r/2 > 1 - r$.

In the case when the initial run of the algorithm from Theorem 40 reports NO, we just apply the algorithm from Theorem 43, again with $p = r/2$. With probability at least $1 - r/2$ the answer NO of the algorithm from Theorem 40 is correct. Conditioned on that, we know that $\mathrm{OPT} > d \geq \frac{122 \ln n}{\epsilon^2}$ and then the algorithm from Theorem 43 returns a $(1 + \epsilon)$-approximation with probability at least $1 - r/2$. Thus, the answer is correct with probability at least $(1 - r/2)^2 > 1 - r$.

The total running time can be bounded as follows

$$\mathcal{O}^\star\left(\left(\frac{3}{\epsilon}\right)^{\frac{244 \ln n}{\epsilon^2}}\right) \subseteq \mathcal{O}^\star\left(n^{\mathcal{O}\left(\frac{\ln 1/\epsilon}{\epsilon^2}\right)}\right) \subseteq n^{\mathcal{O}\left(\frac{\log 1/\epsilon}{\epsilon^2}\right)} \cdot \mathrm{poly}(m).$$

□

Chapter 5

Concluding Remarks and Open Questions

In this thesis we have shown new polynomial-time constant-factor approximation algorithms for a few NP-hard optimization problems that model real-world issues such as clustering and multiwinner elections.

In Chapter 2 we have shown the first constant-factor approximation algorithm for the Ordered k-Median problem. This was achieved by adopting the less detailed version of the analysis of the algorithm by Charikar and Li for k-Median [48] , and hence our constants can probably be improved.

Soon after the submission of our paper [35, 36], Chakrabarty and Swamy [41, 43] announced $18+\epsilon$ and $8.5+\epsilon$ approximation algorithms for Ordered k-Median and Rectangular Ordered k-Median respectively. A few months later Chakrabarty and Swamy [42, 44] improved an approximation constant for Ordered k-Median to $5+\epsilon$. This is quite low constant but still there is a gap to known hardness of approximation constant. Indeed, $(2-\epsilon)$-approximation algorithm for Ordered k-Median would imply $\mathsf{P}=\mathsf{NP}$ due to hardness of k-Center [82].

It is a challenging open problem to close approximability gaps either for k-Median or for Ordered k-Median. Especially, it would be interesting to show hardness of approximation for Ordered k-Median with a constant above currently best known approximation factor for k-Median, i.e., $2.675+\epsilon$ [31].

It might be interesting to see if our methods can be used for other problems with ordered objectives. In particular, relaxing the assumption on weights being non-increasing appears to be a natural direction for future work. Indeed, to see the expressive power of the ordered objective function observe that the k-Median with Outliers problem [47], which is a version of k-Median where the objective function does not pay for the p most expensive connection costs for some fixed p, can naturally be encoded as a version of Ordered k-Median with *increasing* weights. Specifically, k-Median with Outliers is equivalent to Ordered k-Median with a sequence of weights equal to p zeros on the lowest indices and $n-p$ of ones on the highest indices. For k-Median with Outliers a constant factor approximation using local search and Lagrangian preserving multiplier property is known [49].

In Chapter 3 we have introduced a new family of clustering problems, called OWA

k-Median, and we have shown that our problem with the harmonic sequence of weights allows for a constant factor approximation even for general (non-metric) costs. Hence this algorithm applies to Proportional Approval Voting as well. In the analysis of our approximation algorithm for the Harmonic k-Median problem, we used the fact that the dependent rounding procedure satisfies the Binary Negative Association. Also we showed that Metric OWA k-Median can be approximated within a factor of 93 via a reduction to Fault Tolerant k-Median with Clients Multiplicities.

It has been shown that OWA k-Median with p-geometric weights with $p < 1/e$ cannot be approximated without the assumption of the costs being metric [32]. The status of the non-metric problem with p-geometric weights with $p > 1/e$ remains an intriguing open problem. Another interesting direction is to study the computational complexity of the problem with p-geometric weights in a metric space.

Finally, we believe that it is an important future direction to establish some lower bounds for the approximability of the studied problems, in particular for the problem of finding winners under Proportional Approval Voting.

In Chapter 4 we have shown two PTASes for the Minimax Approval Voting problem improving the previous best 2-approximation [39] and we have constructed a parameterized approximation scheme for Minimax Approval Voting that can be of independent interest. Although the asymptotic worst-case complexity of a PTAS from Theorem 43 is better than in the case of a PTAS from Theorem 39, the large constants hidden in the exponents of the function describing the running time still make it far from being practical. A further algorithm engineering research effort can help to turn our ideas into a useful implementation.

There are some unanswered questions related to Minimax Approval Voting. Our PTASes are randomized, and it seems there is no direct way of derandomizing them. It might be interesting to find an equally fast deterministic PTAS. The second question is whether there are even faster PTASes for Closest String or Minimax Approval Voting. Recently, Cygan et al. [57] showed that under ETH, there is no PTAS in time $f(\epsilon)\cdot n^{o(1/\epsilon)}$ for Closest String. This extends to the same lower bound for Minimax Approval Voting, since we can try all values $k \in \{0, 1, \ldots, m\}$. It would be interesting to close the gap in the running time of a PTAS either for Closest String or for Minimax Approval Voting.

Concluding, we believe that the ideas and algorithm analysis techniques developed in this thesis will be useful in further work on approximation algorithms. Also we hope our results will stimulate more interdisciplinary research on relations between clustering problems and multiwinner elections.

The last comment is on the practicality of the presented results. There are examples of theoretical work which was next turned into practical software by means of a non-trivial algorithm engineering effort. See, e.g., the algorithm of Tamaki [144] based on the work of Bouchitte and Todinca [23], which solved all 100 instances of exact treewidth challenge at the PACE 2017 competition [58]. Similarly, we believe that our techniques, possibly augmented with additional ideas, may be used in an efficient implementation. However, known hardness and lower bounds show obstacles which any such implementation has to face.

Bibliography

[1] S. Ahmadian, A. Norouzi-Fard, O. Svensson, and J. Ward. Better Guarantees for k-Means and Euclidean k-Median by Primal-Dual Algorithms. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science (FOCS 2017)*, pages 61–72, 2017. 8, 14

[2] N. Ailon, M. Charikar, and A. Newman. Aggregating Inconsistent Information: Ranking and Clustering. *Journal of the ACM*, 55(5):23:1–23:27, 2008. 3

[3] S. Alamdari and D. B. Shmoys. A Bicriteria Approximation Algorithm for the k-Center and k-Median Problems. In *Proceedings of the 15th International Workshop on Approximation and Online Algorithms (WAOA 1017)*, pages 66–75, 2017. 9, 10

[4] E. Alpaydin. *Introduction to Machine Learning*. MIT Press, 3rd edition, 2014. 4

[5] G. Amanatidis, E. Markakis, and K. Sornat. Inequity Aversion Pricing over Social Networks: Approximation Algorithms and Hardness Results. In *Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, pages 9:1–9:13, 2016. 22

[6] A. Andoni, P. Indyk, and M. Patrascu. On the Optimality of the Dimensionality Reduction Method. In *Proceedings of the 47th IEEE Symposium on Foundations of Computer Science (FOCS 2006)*, pages 449–458, 2006. 19

[7] A. Aouad and D. Segev. The Ordered k-Median Problem: Surrogate Models and Approximation Algorithms. *Mathematical Programming*, 2018. vii, ix, 5, 7, 8, 9, 10, 23, 38, 39

[8] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. 1

[9] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local Search Heuristics for k-Median and Facility Location Problems. *SIAM Journal on Computing*, 33(3):544–562, 2004. 7, 8

[10] A. Auger and B. Doerr. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. World Scientific Publishing, 2011. 49

[11] H. Aziz, A. Bogomolnaia, and H. Moulin. Fair Mixing: The Case of Dichotomous Preferences. *CoRR*, abs/1712.02542, 2017. 4

[12] H. Aziz, M. Brill, V. Conitzer, E. Elkind, R. Freeman, and T. Walsh. Justified Representation in Approval-Based Committee Voting. *Social Choice and Welfare*, 48(2):461–485, 2017. 3, 4, 13, 17

[13] H. Aziz, S. Gaspers, J. Gudmundsson, S. Mackenzie, N. Mattei, and T. Walsh. Computational Aspects of Multi-Winner Approval Voting. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, pages 107–115, 2015. 16

[14] M. L. Balinski. On Finding Integer Solutions to Linear Programs. In *Proceedings of the IBM Scientific Computing Symposium on Combinatorial Problems*, pages 225–248, 1966. 2, 3

[15] Y. Bartal. On Approximating Arbitrary Metrices by Tree Metrics. In *Proceedings of the 30th ACM Symposium on Theory of Computing (STOC 1998)*, pages 161–168, 1998. 7

[16] D. Baumeister, T. Bohnlein, L. Rey, O. Schaudt, and A. Selker. Minisum and Minimax Committee Election Rules for General Preference Types. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*, pages 1656–1657, 2016. 19

[17] D. Bertsimas and R. Mazumder. Least Quantile Regression via Modern Optimization. *The Annals of Statistics*, 42(6):2494–2525, 12 2014. 7

[18] D. Bertsimas and M. Sim. Robust Discrete Optimization and Network Flows. *Mathematical Programming*, 98(1-3):49–71, 2003. 7

[19] D. Bertsimas and R. Weismantel. *Optimization Over Integers*. Athena Scientific, 2005. 7

[20] N. Betzler, A. Slinko, and J. Uhlmann. On the Computation of Fully Proportional Representation. *Journal of Artificial Intelligence Research*, 47:475–519, 2013. 17

[21] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958. 3

[22] N. Boland, P. Domínguez-Marín, S. Nickel, and J. Puerto. Exact Procedures for Solving the Discrete Ordered Median Problem. *Computers & OR*, 33(11):3270–3300, 2006. 8

[23] V. Bouchitte and I. Todinca. Treewidth and Minimum Fill-in: Grouping the Minimal Separators. *SIAM Journal on Computing*, 31(1):212–232, 2001. 86

[24] G. E. P. Box. Robustness in the Strategy of Scientific Model Building. In R. L. Launer and G. N. Wilkinson, editors, *Robustness in Statistics*, pages 201–236. Academic Press, 1979. 1

[25] P. S. Bradley, U. M. Fayyad, and O. L. Mangasarian. Mathematical Programming for Data Mining: Formulations and Challenges. *INFORMS Journal on Computing*, 11(3):217–238, 1999. 7

[26] S. J. Brams and P. C. Fishburn. *Approval Voting*. Springer, 2nd edition, 2007. 16

[27] S. J. Brams, D. M. Kilgour, and M. R. Sanver. A Minimax Procedure for Negotiating Multilateral Treaties. In R. Avenhaus and I. W. Zartman, editors, *Diplomacy Games*, pages 265–282. Springer, 2007. 16

[28] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016. 4

[29] R. Bredereck, J. Chen, P. Faliszewski, J. Guo, R. Niedermeier, and G. J. Woeginger. Parameterized Algorithmics for Computational Social Choice: Nine Research Challenges. *Tsinghua Science and Technology*, 19(4):358–373, Aug 2014. 18

[30] M. Brill, J. Laslier, and P. Skowron. Multiwinner Approval Rules as Apportionment Methods. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017)*, pages 414–420, 2017. 13

[31] J. Byrka, T. Pensyl, B. Rybicki, A. Srinivasan, and K. Trinh. An Improved Approximation for k-Median and Positive Correlation in Budgeted Optimization. *ACM Transactions on Algorithms*, 13(2):23:1–23:31, 2017. 7, 8, 14, 85

[32] J. Byrka, P. Skowron, and K. Sornat. Proportional Approval Voting, Harmonic k-Median, and Negative Association. *CoRR*, abs/1704.02183, 2017. 16, 86

[33] J. Byrka, P. Skowron, and K. Sornat. Proportional Approval Voting, Harmonic k-Median, and Negative Association. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, pages 26:1–26:14, 2018. 22

[34] J. Byrka and K. Sornat. PTAS for Minimax Approval Voting. In *Proceedings of the 10th Conference on Web and Internet Economics (WINE 2014)*, pages 203–217, 2014. 22

[35] J. Byrka, K. Sornat, and J. Spoerhase. Constant-Factor Approximation for Ordered k-Median. *CoRR*, abs/1711.01972, 2017. 11, 85

[36] J. Byrka, K. Sornat, and J. Spoerhase. Constant-Factor Approximation for Ordered k-Median. In *Proceedings of the 50th ACM Symposium on Theory of Computing (STOC 2018)*, pages 620–631, 2018. 11, 22, 85

[37] I. Caragiannis, J. A. Covey, M. Feldman, C. M. Homan, C. Kaklamanis, N. Karanikolas, A. D. Procaccia, and J. S. Rosenschein. On the Approximability of Dodgson and Young Elections. *Artificial Intelligence*, 187:31–51, 2012. 3, 4

[38] I. Caragiannis, C. Kaklamanis, N. Karanikolas, and A. D. Procaccia. Socially Desirable Approximations for Dodgson's Voting Rule. *ACM Transactions on Algorithms*, 10(2):6:1–6:28, 2014. 3, 4

[39] I. Caragiannis, D. Kalaitzis, and E. Markakis. Approximation Algorithms and Mechanism Design for Minimax Approval Voting. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010)*, pages 737–742, 2010. vii, ix, 5, 17, 18, 19, 65, 81, 86

[40] J. Carlson, A. Jaffe, and A. Wiles, editors. *The Millennium Prize Problems*. American Mathematical Society, 2006. 1

[41] D. Chakrabarty and C. Swamy. Interpolating Between k-Median and k-Center: Approximation Algorithms for Ordered k-Median. *CoRR*, abs/1711.08715, 2017. 11, 85

[42] D. Chakrabarty and C. Swamy. Approximation Algorithms for Minimum Norm and Ordered Optimization Problems. *CoRR*, abs/1811.05022, 2018. 11, 85

[43] D. Chakrabarty and C. Swamy. Interpolating between k-Median and k-Center: Approximation Algorithms for Ordered k-Median. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, pages 29:1–29:14, 2018. 11, 85

[44] D. Chakrabarty and C. Swamy. Approximation Algorithms for Minimum Norm and Ordered Optimization Problems. In *Proceedings of the 51th ACM Symposium on Theory of Computing (STOC 2019)*, 2019. 11, 85

[45] J. R. Chamberlin and P. N. Courant. Representative Deliberations and Representative Decisions: Proportional Representation and the Borda Rule. *American Political Science Review*, 77:718–733, 9 1983. 3, 13

[46] M. Charikar, S. Guha, É. Tardos, and D. B. Shmoys. A Constant-Factor Approximation Algorithm for the k-Median Problem. *Journal of Computer and System Sciences*, 65(1):129–149, 2002. 2, 7

[47] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for Facility Location Problems with Outliers. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2001)*, pages 642–651, 2001. 85

[48] M. Charikar and S. Li. A Dependent LP-Rounding Approach for the k-Median Problem. In *Proceedings of the 39th International Colloquium on Automata, Languages, and Programming (ICALP 2012)*, pages 194–205, 2012. 8, 9, 11, 24, 25, 26, 27, 29, 31, 32, 85

[49] K. Chen. A Constant Factor Approximation Algorithm for k-Median Clustering with Outliers. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2008)*, pages 826–835, 2008. 85

[50] V. Conitzer. Making Decisions Based on the Preferences of Multiple Agents. *Communications of the ACM*, 53(3):84–94, 2010. 3

[51] S. A. Cook. The Complexity of Theorem-Proving Procedures. In *Proceedings of the 3rd ACM Symposium on Theory of Computing (STOC 1971)*, pages 151–158, 1971. 1

[52] D. Coppersmith, L. Fleischer, and A. Rudra. Ordering by Weighted Number of Wins Gives a Good Ranking for Weighted Tournaments. *ACM Transactions on Algorithms*, 6(3):55:1–55:13, 2010. 3

[53] G. Cornuéjols, G. L. Nemhauser, and L. A. Wolsey. The Uncapacitated Facility Location Problem. In P. B. Mirchandani and R. L. Francis, editors, *Discrete Location Theory*, pages 119–171. John Wiley and Sons, 1990. 3

[54] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015. 18

[55] M. Cygan, Ł. Kowalik, A. Socała, and K. Sornat. Approximation and Parameterized Complexity of Minimax Approval Voting. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017)*, pages 459–465, 2017. 22

[56] M. Cygan, L. Kowalik, A. Socała, and K. Sornat. Approximation and Parameterized Complexity of Minimax Approval Voting. *Journal of Artificial Intelligence Research*, 63:495–513, 2018. vii, ix, 5, 20, 21, 22

[57] M. Cygan, D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. Lower Bounds for Approximation Schemes for Closest String. In *Proceedings of the 15th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2016)*, pages 12:1–12:10, 2016. 86

[58] H. Dell, C. Komusiewicz, N. Talmon, and M. Weller. The PACE 2017 Parameterized Algorithms and Computational Experiments Challenge: The Second Iteration. In *Proceedings of the 12th International Symposium on Parameterized and Exact Computation (IPEC 2017)*, pages 30:1–30:12, 2017. 86

[59] C. Dodgson. *A Method of Taking Votes on More Than Two Issues*. Pamphlet printed by the Clarendon Press, Oxford, and headed "not yet published", 1876. 3

[60] P. Domínguez-Marín, S. Nickel, P. Hansen, and N. Mladenovic. Heuristic Procedures for Solving the Discrete Ordered Median Problem. *Annals OR*, 136(1):145–173, 2005. 8

[61] Z. Drezner and S. Nickel. Constructing a DC Decomposition for Ordered Median Problems. *Journal of Global Optimization*, 45(2):187–201, 2009. 8

[62] Z. Drezner and S. Nickel. Solving the Ordered One-Median Problem in the Plane. *European Journal of Operational Research*, 195(1):46–61, 2009. 8

[63] D. P. Dubhashi, J.Jonasson, and D. Ranjan. Positive Influence and Negative Dependence. *Combinatorics, Probability & Computing*, 16(1):29–41, 2007. 46, 56

[64] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank Aggregation Methods for the Web. In *Proceedings of the 10th International World Wide Web Conference (WWW 2001)*, pages 613–622, 2001. 3

[65] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. Properties of Multiwinner Voting Rules. *Social Choice and Welfare*, 48(3):599–632, 2017. 4, 17

[66] U. Endriss, editor. *Trends in Computational Social Choice*. AI Access, 2017. 4

[67] I. Espejo, A. M. Rodríguez-Chía, and C. Valero. Convex Ordered Median Problem with ℓ_p-Norms. *Computers & OR*, 36(7):2250–2262, 2009. 8

[68] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Multimode Control Attacks on Elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011. 3

[69] P. Faliszewski, P. Manurangsi, and K. Sornat. Approximation and Hardness of Shift-Bribery. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, to appear. 22

[70] P. Faliszewski, J. Sawicki, R. Schaefer, and M. Smolka. Multiwinner Voting in Genetic Algorithms for Solving Ill-Posed Global Optimization Problems. In *Proceedings of the 19th European Conference on the Applications of Evolutionary Computation (EvoApplications 2016)*, pages 409–424, 2016. 3

[71] P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. Multiwinner Voting: A New Challenge for Social Choice Theory. In U. Endriss, editor, *Trends in Computational Social Choice*, chapter 2, pages 27–47. AI Access, 2017. 17

[72] R. Z. Farahani and M. Hekmatfar. *Facility Location: Concepts, Models, Algorithms and Case Studies*. Contributions to Management Science. Springer, 2009. 2

[73] E. Fernández, M. A. Pozo, J. Puerto, and A. Scozzari. Ordered Weighted Average Optimization in Multiobjective Spanning Tree Problem. *European Journal of Operational Research*, 260(3):886–903, 2017. 7

[74] P. C. Fishburn. Axioms for Approval Voting: Direct Proof. *Journal of Economic Theory*, 19(1):180–185, 1978. 16

[75] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent Rounding and Its Applications to Approximation Algorithms. *Journal of the ACM*, 53(3):324–360, 2006. 15, 46

[76] W. I. Gasarch. Guest Column: The Third P=?NP Poll. *SIGACT News*, 50(1):38–59, 2019. 1

[77] J. Gramm, R. Niedermeier, and P. Rossmanith. Fixed-Parameter Algorithms for Closest String and Related Problems. *Algorithmica*, 37(1):25–42, 2003. 19

[78] M. T. Hajiaghayi, W. Hu, J. Li, S. Li, and B. Saha. A Constant Factor Approximation Algorithm for Fault-Tolerant k-Median. *ACM Transactions on Algorithms*, 12(3):36:1–36:19, 2016. 14, 16, 45, 60, 61

[79] S. L. Hakimi. Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph. *Operations Research*, 12(3):450–459, 1964. 2, 13

[80] S. L. Hakimi. Optimum Distribution of Switching Centers in a Communication Network and Some Related Graph Theoretic Problems. *Operations Research*, 13(3):462–475, 1965. 2, 13

[81] D. G. Harris, T. Pensyl, A. Srinivasan, and K. Trinh. Symmetric Randomized Dependent Rounding. *CoRR*, abs/1709.06995v1, 2017. 9

[82] D. Hochbaum and D. Shmoys. A Best Possible Heuristic for the k-Center Problem. *Mathematics of Operations Research*, 10(2):180–184, 1985. 2, 6, 7, 8, 14, 19, 85

[83] C. M. Homan and L. A. Hemaspaandra. Guarantees for the Success Frequency of an Algorithm for Finding Dodgson-Election Winners. *Journal of Heuristics*, 15(4):403–423, 2009. 3

[84] R. Impagliazzo and R. Paturi. On the Complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. 19, 20

[85] A. K. Jain. Data Clustering: 50 Years Beyond k-Means. *Pattern Recognition Letters*, 31(8):651–666, 2010. 2

[86] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. Greedy Facility Location Algorithms Analyzed Using Dual Fitting with Factor-Revealing LP. *Journal of the ACM*, 50(6):795–824, 2003. 7, 8

[87] K. Joag-Dev and F. Proschan. Negative Association of Random Variables with Applications. *The Annals of Statistics*, 11(1):286–295, 1983. 16, 46

[88] J. Kalcsics, S. Nickel, and J. Puerto. Multifacility Ordered Median Problems on Networks: A Further Analysis. *Networks*, 41(1):1–12, 2003. 8

[89] J. Kalcsics, S. Nickel, J. Puerto, and A. Tamir. Algorithmic Results for Ordered Median Problems. *Operations Research Letters*, 30(3):149–158, 2002. 8

[90] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A Local Search Approximation Algorithm For k-Means Clustering. *Computational Geometry*, 28(2-3):89–112, 2004. 8

[91] O. Kariv and S. L. Hakimi. An Algorithmic Approach to Network Location Problems. II: The p-Medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979. 2, 13

[92] N. Karmarkar. A New Polynomial-Time Algorithm for Linear Programming. *Combinatorica*, 4(4):373–396, 1984. 81

[93] R. M. Karp. Reducibility Among Combinatorial Problems. In *Proceedings of a Symposium on the Complexity of Computer Computations*, pages 85–103, 1972. 1

[94] J. G. Kemeny. Mathematics Without Numbers. *Daedalus*, 88(4):577–591, 1959. 3

[95] M. G. Kendall. *Rank Correlation Methods*. Griffin, London, 1948. 3

[96] C. Kenyon-Mathieu and W. Schudy. How to Rank with Few Errors. In *Proceedings of the 39th ACM Symposium on Theory of Computing (STOC 2007)*, pages 95–103, 2007. 3

[97] D. M. Kilgour. Approval Balloting for Multi-Winner Elections. In J.-F. Laslier and R. M. Sanver, editors, *Handbook on Approval Voting*, pages 105–124. Springer, 2010. 14, 16, 17

[98] J. M. Kleinberg, C. H. Papadimitriou, and P. Raghavan. Segmentation Problems. *Journal of the ACM*, 51(2):263–280, 2004. 3

[99] J. B. Kramer, J. Cutler, and A. J. Radcliffe. Negative Dependence and Srinivasan's Sampling Process. *Combinatorics, Probability & Computing*, 20(3):347–361, 2011. 46, 49, 56

[100] M. Labbé, D. Ponce, and J. Puerto. A Comparative Study of Formulations and Solution Methods for the Discrete Ordered p-Median Problem. *Computers & OR*, 78:230–242, 2017. 8

[101] G. Laporte, S. Nickel, and F. Saldanha da Gama, editors. *Location Science*. Springer, 2015. 7

[102] J.-F. Laslier and R. M. Sanver. *Handbook on Approval Voting*. Studies in Choice and Welfare. Springer, 2010. 16

[103] R. LeGrand. Analysis of the Minimax Procedure. Technical Report WUCSE-2004-67, Department of Computer Science and Engineering, Washington University, St. Louis, Missouri, 2004. 17, 18

[104] R. LeGrand, E. Markakis, and A. Mehta. Some Results on Approximating the Minimax Solution in Approval Voting. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, pages 1193–1195, 2007. 18

[105] L. A. Levin. Universal Sequential Search Problems. *Problems of Information Transmission*, 9(3):265–266, 1973. 1

[106] M. Li, B. Ma, and L. Wang. On the Closest String and Substring Problems. *Journal of the ACM*, 49(2):157–171, 2002. 17, 19, 20

[107] S. Li and O. Svensson. Approximating k-Median via Pseudo-Approximation. *SIAM Journal on Computing*, 45(2):530–547, 2016. 7, 8

[108] H. Liu and J. Guo. Parameterized Complexity of Winner Determination in Minimax Committee Elections. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016)*, pages 341–349, 2016. 19

[109] D. Lokshtanov, D. Marx, and S. Saurabh. Lower Bounds Based on the Exponential Time Hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011. 20

[110] D. Lokshtanov, D. Marx, and S. Saurabh. Slightly Superexponential Parameterized Problems. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2011)*, pages 760–776, 2011. 19

[111] T. Lu and C. Boutilier. Budgeted Social Choice: From Consensus to Personalized Decision Making. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 280–286, 2011. 3

[112] T. Lu and C. Boutilier. Value-Directed Compression of Large-Scale Assignment Problems. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, pages 1182–1190, 2015. 3

[113] B. Ma and X. Sun. More Efficient Algorithms for Closest String and Substring Problems. *SIAM Journal on Computing*, 39(4):1432–1443, 2009. 19

[114] J. C. McCabe-Dansted, G. Pritchard, and A. M. Slinko. Approximability of Dodgson's Rule. *Social Choice and Welfare*, 31(2):311–330, 2008. 3

[115] N. Misra. personal communication, 2016. 19

[116] N. Misra, A. Nabeel, and H. Singh. On the Parameterized Complexity of Minimax Approval Voting. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, pages 97–105, 2015. 18, 19, 20, 80

[117] B. L. Monroe. Fully Proportional Representation. *American Political Science Review*, 89(4):925–940, 1995. 3

[118] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995. 66

[119] S. Nickel. Discrete Ordered Weber Problems. In *Operations Research Proceedings (SOR 2001)*, pages 71–76, 2001. 7, 8

[120] S. Nickel and J. Puerto. A Unified Approach to Network Location Problems. *Networks*, 34(4):283–290, 1999. 7, 8

[121] S. Nickel and J. Puerto. *Location Theory: A Unified Approach*. Springer, 2009. 7, 8

[122] R. Niedermeier. Lower Bound Issues in Computational Social Choice. A talk at the workshop Satisfiability Lower Bounds and Tight Results for Parameterized and Exponential-Time Algorithms, Simons Institute, Berkeley, November 10, 2015. 20

[123] M. Pál, É. Tardos, and T. Wexler. Facility Location with Nonuniform Hard Capacities. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS 2001)*, pages 329–338, 2001. 3

[124] J. Puerto and F. R. Fernández. Geometrical Properties of the Symmetrical Single Facility Location Problem. *Journal of Nonlinear and Convex Analysis*, 1(3):321–342, 2000. 7

[125] J. Puerto, D. Pérez-Brito, and C. G. García-González. A Modified Variable Neighborhood Search for the Discrete Ordered Median Problem. *European Journal of Operational Research*, 234(1):61–76, 2014. 8

[126] J. Puerto, A. B. Ramos, and A. M. Rodríguez-Chía. A Specialized Branch & Bound & Cut for Single-Allocation Ordered Median Hub Location Problems. *Discrete Applied Mathematics*, 161(16-17):2624–2646, 2013. 8

[127] J. Puerto and A. M. Rodríguez-Chía. On the Exponential Cardinality of FDS for the Ordered p-Median Problem. *Operations Research Letters*, 33(6):641–651, 2005. 8

[128] J. Puerto, A. M. Rodríguez-Chía, and A. Tamir. Minimax Regret Single-Facility Ordered Median Location Problems on Networks. *INFORMS Journal on Computing*, 21(1):77–87, 2009. 8

[129] J. Puerto, A. M. Rodríguez-Chía, and A. Tamir. Revisiting k-Sum Optimization. *Mathematical Programming*, 165(2):579–604, 2017. 7, 8

[130] J. Puerto and A. Tamir. Locating Tree-Shaped Facilities Using the Ordered Median Objective. *Mathematical Programming*, 102(2):313–338, 2005. 7, 8

[131] J. Rawls. *A Theory of Justice*. Harvard Univeristy Press, revised edition, 1971. 17

[132] A. M. Rodríguez-Chía, J. Puerto, D. Pérez-Brito, and J. A. Moreno. The p-Facility Ordered Median Problem on Networks. *Top*, 13(1):105–126, Jun 2005. 8

[133] L. Sanchez-Fernandez and J. A. Fisteus. Monotonicity Axioms in Approval-Based Multi-Winner Voting Rules. *CoRR*, abs/1710.04246, 2017. 17

[134] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003. 4, 68

[135] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation Algorithms for Facility Location Problems (Extended Abstract). In *Proceedings of the 29th ACM Symposium on Theory of Computing (STOC 1997)*, pages 265–274, 1997. 3

[136] P. Skowron, P. Faliszewski, and J. Lang. Finding a Collective Set of Items: From Proportional Multirepresentation to Group Recommendation. *Artificial Intelligence*, 241:191–216, 2016. 3, 13

[137] P. Skowron, P. Faliszewski, and A. M. Slinko. Achieving Fully Proportional Representation: Approximability Results. *Artificial Intelligence*, 222:67–103, 2015. 3, 4, 12

[138] A. Socała. *Lower Bounds Under Strong Complexity Assumptions*. PhD thesis, University of Warsaw, 2017. 20

[139] A. Srinivasan. Distributions on Level-Sets with Applications to Approximation Algorithms. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS 2001)*, pages 588–597, 2001. 15, 45, 46, 48

[140] Z. Stanimirovic, J. Kratica, and D. Dugosija. Genetic Algorithms for Solving the Discrete Ordered Median Problem. *European Journal of Operational Research*, 182(3):983–1001, 2007. 8

[141] H. Steinhaus. Sur la division des corps matériels en parties (in French). *Bulletin de l'Académie Polonaise des Sciences*, IV(12):801–804, 1956. 2

[142] D. M. K. Steven J. Brams and M. R. Sanver. A Minimax Procedure for Electing Committees. *Public Choice*, 132(3-4):401–420, 2007. 17

[143] C. Swamy and D. Shmoys. Fault-Tolerant Facility Location. *ACM Transactions on Algorithms*, 4(4):51:1–51:27, 2008. 14

[144] H. Tamaki. Positive-Instance Driven Dynamic Programming for Treewidth. In *Proceedings of the 25th Annual European Symposium on Algorithms (ESA 2017)*, pages 68:1–68:13, 2017. 86

[145] A. Tamir. The k-Centrum Multi-Facility Location Problem. *Discrete Applied Mathematics*, 109(3):293–307, 2001. 7, 8, 9

[146] A. Tamir, D. Pérez-Brito, and J. A. Moreno-Pérez. A Polynomial Algorithm for the p-Centdian Problem on a Tree. *Networks*, 32(4):255–262, 1998. 7, 9

[147] T. N. Thiele. Om Flerfoldsvalg. In *Oversigt over det Kongelige Danske Videnskabernes Selskabs Forhandlinger (in Danish)*, pages 415–441. København: A.F. Høst, 1895. 3, 13

[148] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011. 2, 6, 8

[149] R. R. Yager. On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183–190, 1988. 11

[150] H. Young. An Axiomatization of Borda's Rule. *Journal of Economic Theory*, 9(1):43–52, 1974. 4