Department of Computer Science

Parameterized Approximation Results for Clustering and Graph Packing Problems

Ameet Gadekar

Aalto University

DOCTORAL THESES

Aalto University publication series
DOCTORAL THESES 227/2023

Parameterized Approximation Results for Clustering and Graph Packing Problems

Ameet Gadekar

A doctoral thesis completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall T3, Konemiehentie 2, of the school on 25 January 2024 at 13.

Aalto University
School of Science
Department of Computer Science
Theoretical Computer Science

Supervising professor
Professor Parinya Chalermsook, Aalto University, Finland

Thesis advisor
Professor Chris Brzuska, Aalto University, Finland

Preliminary examiners
Dr. Chien-Chung Huang, CNRS, Ecole Normale Superieure Ulm, Paris, France
Prof. Dr. habil. Andreas Emil Feldmann, University of Sheffield, United Kingdom

Opponent
Dr. Chien-Chung Huang, CNRS, Ecole Normale Superieure Ulm, Paris, France

Aalto University publication series
DOCTORAL THESES 227/2023

ISBN 978-952-64-1602-1 (printed)
ISBN 978-952-64-1603-8 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-64-1603-8

Unigrafia Oy
Helsinki 2023

Finland

Printed matter
4041-0619

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Ameet Gadekar

Name of the doctoral thesis
Parameterized Approximation Results for Clustering and Graph Packing Problems

Publisher School of Science

Unit Department of Computer Science

Series Aalto University publication series DOCTORAL THESES 227/2023

Field of research Theoretical Computer Science

Manuscript submitted 7 August 2023 **Date of the defence** 25 January 2024

Permission for public defence granted (date) 1 November 2023 **Language** English

☐ **Monograph** ☒ **Article thesis** ☐ **Essay thesis**

Abstract

The domains of clustering and graph packing have been the focus of extensive research across multiple disciplines, including optimization, machine learning, data mining, computational geometry, and operations research. Many central problems in these domains are known to be NP-hard, prompting the exploration of approximation algorithms and parameterized algorithms as possible approaches, among others. In recent times, the paradigm of parameterized approximation algorithms, which strike a balance between approximation and polynomial-time computability on instances with small parameters, has gained renewed attention.

This thesis comprises two distinct parts. In Part I, we design parameterized approximation algorithms for several clustering problems, significantly advancing the state of the art. In the center based *k-clustering* problem, which includes the classical problems of *k-median, k-means*, and *k-center*, we are given a point set P and the goal is to find a *k*-partition (clusters) of P along with a representative (center) for each partition to minimize certain clustering objective that is a function of the distance vector - the vector of distances between the centers and the points in the corresponding clusters. In the *Norm k-Clustering* problem, the clustering objective is a monotone norm of the distance vector. This objective is quite general and captures essentially all the clustering objectives studied so far. In this thesis, we design a novel and simple *Efficient Parameterized Approximation Schemes (EPAS)* framework for *Norm k-Clustering* in several metric spaces. This result unifies several existing EPASes that are known to be conceptually different. Moreover, our framework resolves many of the open problems related to advanced objectives, including modern constraints on fairness, robustness, and diversity. A notable contribution of this work is a new combinatorial measure of a metric space, which we call *Scatter Dimension*, that enables designing EPAS that is oblivious to the underlying metric space. Additionally, we address other clustering problems, namely *Robust (k-z)-Clustering* and *Diversity-aware k-Median,* and design tight parameterized approximation algorithms for them.

Part II adopts a complementary approach, focusing on lower bounds for graph packing problems. In particular, we consider a notoriously hard problem called *Set Packing* and establish a parameterized dichotomy for the problem. A novel conceptual contribution to this dichotomy is the notion *of compact instances*, which remain challenging to solve despite their small size. Furthermore, we explore the connection between the approximating maximum independent set problem in *k-claw-free graphs* and several convex relaxations.

Keywords clustering, graph packing, parameterized complexity, approximation algorithms

ISBN (printed) 978-952-64-1602-1 **ISBN (pdf)** 978-952-64-1603-8

ISSN (printed) 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2023

Pages 225 **urn** http://urn.fi/URN:ISBN:978-952-64-1603-8

Preface

It has been an adventurous six years since I embarked on my PhD journey! Undeniably, it has been a roller coaster ride of ups and downs, and the thing with roller coasters is that *downs* are more thrilling than *ups*. Looking back, there were moments when self-doubt tried to overshadow my abilities and perseverance, casting shadows on my ambition to complete this thesis. Nevertheless, this journey has not only shaped me into a researcher, but it has also transformed me into a different and improved version of myself. As I reflect on this remarkable journey, I realize that its success is attributed to the support and contributions of many individuals.

Without a doubt, my supervisor, Prof. Parinya Chalermsook, stands out as the first person whose unwavering support and encouragement have been a constant source of motivation, propelling me forward even during the toughest times. I cannot emphasize enough how much his belief in me, especially when I doubted myself, has meant to me. His generous funding enabled me to attend important events and connect with fellow researchers. In every possible way, Parinya has been instrumental in my growth as a researcher and as an individual. I am genuinely grateful to have him as my supervisor, and I thank him for everything!

I am immensely grateful to my co-supervisor, Prof. Chris Brzuska, for being an essential part of this journey. His Advanced Cryptography course in the Math Department sparked my fascination with complexity theoretic cryptography. I have had the pleasure of organizing several reading groups with Chris and his group, and those discussions have been truly memorable. His thirst for knowledge and willingness to explore new ideas have continually amazed me and inspired my own academic pursuits. Beyond academia, Chris has been a good friend, providing support and encouragement during both good and challenging times.

The seeds of my journey were sown during my master's at the Indian Institute of Science, Bangalore. Prof. Arnab Bhattacharyya, my then-advisor, introduced me to the beautiful realm of theoretical computer science. I am profoundly grateful to Arnab for guiding me into this field and nurturing my interest.

I am grateful to the Helsinki Doctoral Education Network in Information and Communications Technology for their doctoral scholarship that supported the first four years of my PhD. I also thank the Department of Computer Science at Aalto University for their funding and making my stay enjoyable. I thank Jarek Byrka for inviting us for a research visit to Wrocław. I would like to thank Danupon Nanongkai, scientific director at Max Planck Institute for Informatics in Saarbrücken, Germany, for partially supporting my research visit to MPII, Saarbrücken, in the winter of 2022. This visit has been very fruitful. I am also thankful to the Universidad de Chile for partially supporting me to attend a workshop on approximation algorithms in Chile in 2019. I am grateful to Dagstuhl Seminar for inviting me to the workshop on Parameterized Approximation. It was a fun and inspirational experience to talk to experts in the field.

I am deeply grateful to my collaborators, as without them, this thesis would have remained a distant dream. I extend my heartfelt thanks to Joachim Spoerhase and Kamyar Khodamoradi for their valuable contributions and friendship. Working with them has been a joyful experience, both professionally and personally. I am thankful for the fruitful collaborations with Suhas Thejaswi and Bruno Ordozgoiti and the many cherished moments we shared. I feel fortunate to have had the opportunity to work with many brilliant researchers, including (but not limited to) Dániel Marx, Jarosław Byrka, Roohani Sharma, Fateme Abbasi, Sandip Banerjee, Pihla Karanko, Cigdem Aslay, and Michał Osadnik. I extend my gratitude to each of them for their valuable contributions.

I would like to express my gratitude to the pre-examiners, Prof. Dr. Andreas Emil Feldmann and Dr. Chien-Chung Huang, for their invaluable contributions and thoughtful feedback on the thesis. Additionally, I express my thanks to Dr. Chien-Chung Huang for graciously agreeing to serve as the opponent.

My friends have been a constant source of rejuvenation and support. Thanks to Nidia, Suhas, and Kunal for making me experience beautiful things outside of work. I thank Sumedha, Manu, and Apurva for being part of my life; they have felt like family to me. My labmates have been an integral part of this journey. I thank Po, Sumedha, Joachim, Sorrachai, Nidia, Marga, Gorav, Ly, Hasti, Michał, Kamyar, Kirthi, Minoo, Natalia, Zahra, and Felix for being around and keeping up the spirit.

Lastly, I owe a debt of gratitude to my family. My parents, Ranjana and Bhujang, have been unwavering in their support and belief in me, which has been a driving force behind my success. They have been a constant source of inspiration throughout my life. I am thankful for my brother, Pankaj, and sister, Nikita, whose presence in my life has brought joy and comfort. To my wife, Smita, I express my heartfelt appreciation for her understanding and constant support. And to my son, Arunoday, you bring happiness to my life every single day, and I am grateful for the love you

bring into our family.

As I come to the end of this journey, I am aware that there may be many individuals whom I have unintentionally omitted, yet their presence and influence in my journey have been significant. To each and every one of them, I extend my heartfelt gratitude for being a part of my life.

Espoo, December 12, 2023,

Ameet Gadekar

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Fateme Abbasi, Sandip Banerjee, Jarosław Byrka, Parinya Chalermsook, **Ameet Gadekar**, Kamyar Khodamoardi, Dániel Marx, Roohani Sharma, Joachim Spoerhase. Parameterized Approximation Schemes for Clustering with General Norm Objectives. Accepted for publication in *64th IEEE Symposium on Foundations of Computer Science (FOCS)*, November 2023.

II Fateme Abbasi, Sandip Banerjee, Jarosław Byrka, Parinya Chalermsook, **Ameet Gadekar**, Kamyar Khodamoardi, Dániel Marx, Roohani Sharma, Joachim Spoerhase. Parameterized Approximation for Robust Clustering in Discrete Geometric Spaces. *Submitted*, August 2023.

III Suhas Thejaswi, **Ameet Gadekar**, Bruno Ordozgoiti, Michał Osadnik. Clustering with Fair-Center Representation: Parameterized Approximation Algorithms and Heuristics. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Pages 1749–1759, August 2022.

IV **Ameet Gadekar**. On the Parameterized Complexity of Compact Set Packing. In *Proceedings of the 17th International Conference and Workshops on Algorithms and Computation (WALCOM)*, Springer Nature Switzerland, Pages 359–370, March 2023.

V Parinya Chalermsook, **Ameet Gadekar**, Kamyar Khodamoardi, Joachim Spoerhase. Independent set in k-Claw-Free Graphs: Conditional χ-boundedness and the Power of LP/SDP Relaxations. Accepted for publication in *The Workshop on Approximation and Online Algorithms (WAOA)*, September 2023.

Author's Contribution

Publication I: "Parameterized Approximation Schemes for Clustering with General Norm Objectives"

The original problem was initially proposed to the author, Kamyar Khodamoradi, and Joachim Spoerhase by Parinya Chalermsook. The author actively participated in this project, engaging in every group brainstorming session. Alongside Parinya Chalermsook, Kamyar Khodamoradi, and Joachim Spoerhase, the author contributed to the ideas, derivation of technical results, and writing of the prior manuscripts, which laid the conceptual and technical foundations for this manuscript. The main part of the conceptual contribution of the paper is due to the intense brainstorming sessions of the author with Parinya Chalermsook, Kamyar Khodamoardi, and Joachim Spoerhase. Other parts of the conceptual contribution are partly due to the research visit of the author with Kamyar Khodamoardi and Joachim Spoerhase to Fateme Abbasi, Sandip Banerjee, Jarosław Byrka, and partly due to the research visit of the author with Parinya Chalermsook, Kamyar Khodamoardi, and Joachim Spoerhase to Dániel Marx and Roohani Sharma. The author also actively participated in the development of the final algorithm of this manuscript, which was mainly due to the ideas of Joachim Spoerhase and Dániel Marx. Additionally, the author provided assistance to Joachim Spoerhase in writing the technical results of this manuscript. The writing of the manuscript was a collaborative effort, with everyone making significant contributions.

Publication II: "Parameterized Approximation for Robust Clustering in Discrete Geometric Spaces"

The problem was initially proposed to the author, Kamyar Khodamoradi, and Joachim Spoerhase by Parinya Chalermsook. Subsequently, the author actively participated in every brainstorming meeting with them. The author independently came up with the ideas and designed the algorithm for the bounded dimensional Euclidean space. Moreover, the author was solely responsible for deriving and writing the corresponding technical results. Additionally, the author actively contributed to the development of the algorithm for the high-dimensional space with everyone. Fateme Abbasi, Sandip Banerjee, and Jarosław Byrka were responsible for deriving the results of the high-dimensional space. Dániel Marx came up with the hardness result. The writing of the manuscript was a collaborative effort.

Publication III: “Clustering with Fair-Center Representation: Parameterized Approximation Algorithms and Heuristics”

The problem was initially proposed to the author by Suhas Thejaswi. Subsequently, the author actively participated in intensive brainstorming sessions with Suhas Thejaswi and Bruno Ordozgoiti. The author took the lead in deriving, formulating, and writing the technical proofs. Implementation and experiments are due to Suhas Thejaswi and Michał Osadnik. The writing of the manuscript was a collaborative effort by the author with Suhas Thejaswi and Bruno Ordozgoiti.

Publication IV: “On the Parameterized Complexity of Compact Set Packing”

The author took full responsibility for all aspects of this article. This included conceiving the problem, brainstorming innovative ideas, formulating and proving the technical theorems, as well as writing the entire manuscript. The author gratefully acknowledges the guidance and support of their advisor, Parinya Chalermsook, who provided valuable input during the presentation of the results.

Publication V: “Independent set in k-Claw-Free Graphs: Conditional χ-boundedness and the Power of LP/SDP Relaxations”

The problem was initially proposed to the author, Kamyar Khodamoradi, and Joachim Spoerhase by Parinya Chalermsook. Subsequently, the author actively participated in every brainstorming meeting with them. The results were a collaborative effort. The author, along with Kamyar Khodamoradi, was primarily responsible for deriving most of the technical results. The writing of the manuscript was a joint effort by the author and Parinya Chalermsook.

List of Figures

1. Introduction

Many central problems in combinatorial optimization are **NP**-hard to solve. In other words, these problems do not admit efficient (polynomial time) algorithms under the widely believed hypothesis of $\mathbf{P} \neq \mathbf{NP}$. Consequently, researchers in the algorithmic community have sought alternative approaches that involve certain trade-offs, which can still be good for practical purposes. Here is a non-exhaustive list of such trade-offs that have garnered significant attention from the community over the past few decades.

Approximation algorithms. One such extensively studied trade-off is the quality of the solution. In this regime, we let go of our ambition of finding an optimal solution and instead settle for a near-optimal one. The algorithms in this approach aim to output a solution in polynomial time that is approximately good, even in the worst-case scenarios. This paradigm enables us to efficiently tackle hard problems and find solutions that are close to optimal in a computationally feasible manner.

Parameterized algorithms. In this regime, which has been studied for more than two decades, we focus on designing efficient algorithms for problems on restricted inputs. This involves identifying a combinatorial property of the input, referred to as the *parameter*, and developing algorithms that are efficient for input instances with a bounded parameter. The running time of these algorithms is known as the parameterized running time. By leveraging the specific structure captured by the parameter, these algorithms can efficiently handle input instances that would otherwise be computationally challenging.

Parameterized Approximation algorithms. This regime, as the name suggests, combines the above two approaches. Although these algorithms have existed for more than 20 years, they have received recent interest from the community. Specifically, here we design approximation algorithms that have parameterized running times. Thus, these algorithms can solve a much broader range of problems than the above two approaches combined, making them more powerful and versatile.

Exponential time algorithms. Another trade-off that is natural to study is the design of fast exponential time algorithms. In this setting, our goal is to solve the problem either exactly or approximately, but with algorithms that have fast exponential running times. This approach allows us to trade efficiency for quality of the solution, enabling us to tackle challenging problems with improved time complexity.

Average case algorithms. In contrast to the approaches we discussed earlier, in this setting, instead of focusing on worst-case inputs, the algorithms are designed to perform well on average or typical instances that are encountered in practical scenarios. These algorithms do not guarantee optimal solutions for all inputs, but they aim to be efficient and provide good results for inputs that are commonly encountered. The motivation behind average-case algorithms is to avoid the overkill of worst-case guarantees when we have knowledge about the distribution of inputs in real-world applications. For example, in the problem of finding the shortest path between two cities, it is often observed that most people search for paths between well-known cities, leading to a non-uniform distribution of user queries. In such cases, it may suffice to have an algorithm that efficiently handles typical instances (commonly occurring scenarios) while being less concerned about its performance on less common inputs.

In addition to designing algorithms to solve these problems efficiently, significant effort has been dedicated to designing lower bounds for them. The objective of this pursuit is to rule out certain types of algorithms, as described above, under standard complexity assumptions. Often, the study of lower bounds and upper bounds complements each other, with researchers striving to find efficient algorithms (upper bounds) while simultaneously exploring the inherent difficulty of the problems through lower bounds. The interplay between lower bounds and upper bounds provides valuable insights into the computational nature of these problems. It aids researchers in identifying the boundaries of computational feasibility, shaping research directions, and addressing unresolved questions in the realm of computational complexity theory.

This thesis explores two broad directions for problems in combinatorial optimization. Firstly, it focuses on designing parameterized approximation algorithms for *clustering* problems which are among the most fundamental problems in combinatorial optimization, including machine learning, data mining, and computational geometry. Despite their fundamental nature, these problems are notoriously hard to solve. Secondly, the thesis delves into designing lower bounds for *graph packing* problems, which are one of the most extensively studied problems in the domain.

1.1 Clustering: Fairness, Robustness, and Beyond

Clustering is among the most fundamental computational problems that arise in optimization, machine learning, data mining, and computational geometry. It has received significant attention from the algorithmic community for over half a century. In the center-based *k-clustering problem*, we are given n data points P in a metric space and are asked to partition P into k *clusters* along with a representative (*center*) for each cluster so as to minimize certain clustering objective that is a function of distances between the points and their centers. Classical clustering problems include k-MEDIAN, k-MEANS, and k-CENTER, with objectives being the sum of distances, the sum of squared distances, and the maximum distance of a point to its center, respectively.

In this thesis, our focus is to design parameterized approximation algorithms for several clustering problems across several metric spaces. These algorithms yield an approximate solution while running in time $f(k)\mathsf{poly}(n)$, for some function f. A special subset of these algorithms is called *Efficient Parameterized Approximation Schemes* (EPAS) that provides a $(1+\epsilon)$ approximate solution running in time $f(k,\epsilon)\mathsf{poly}(n)$, for every $\epsilon > 0$. Such approximation schemes[1] have been prominent in the literature for many classical clustering problems [79, 105, 108, 62, 23, 30, 57]. While past research has been successful in handling classical clustering problems; unfortunately, such theories are insufficient for modern constraints such as fairness, diversity, and robustness. In this thesis, we make substantial progress in our understanding of the computational complexity of these advanced clustering problems.

1.1.1 NORM CLUSTERING: Unifying Clustering Objectives

In fact, in Publication I we design an EPAS framework across several metric spaces for clustering objective that is a *monotone norm* of the distances. This objective (and the corresponding problem called NORM k-CLUSTERING) captures essentially all clustering objectives studied in the literature, from the classical objectives of k-MEDIAN, k-MEANS, k-CENTER to more advanced problems such as ℓ-CENTRUM, ORDERED k-MEDIAN, SOCIALLY FAIR k-MEDIAN (or ROBUST k-MEDIAN), (z,q)-FAIR CLUSTERING, and additionally extends to new objectives that are interesting in their own right. We remark that even for basic objectives of k-MEDIAN, k-MEANS, and k-CENTER, different EPASes are known using different techniques even in the same metric space [14, 90]. Similarly, even for a fixed objective, such as k-MEDIAN, different EPASes are known for dif-

[1]We remark that PTASes (Polynomial Time Approximation Schemes), which are incomparable to EPASes, do not exist for continuous k-means, k-median and k-center [55, 12].

ferent metric spaces that exploit the structure of the underlying metric space [90, 57]. In contrast, our framework for NORM k-CLUSTERING is essentially oblivious to the underlying metric space and yet is able to leverage the inherent structure of the metric space. Hence, it is able to unify several EPASes across several metric spaces, additionally yielding EPASes for modern clustering problems such as SOCIALLY FAIR k-MEDIAN, ORDERED k-MEDIAN, and (z,q)-FAIR CLUSTERING. The existence of EPASes for these problems was unknown before our work. The strength of our framework lies in its simplicity and elegance, as it does not rely on coreset construction. Our main results are encapsulated in the following theorem.

Theorem 1.1.1. *Let f be an efficiently computable monotone norm cost function. Then the k-clustering problem with cost function f admits an EPAS for the following input metrics: (i) metrics of bounded doubling dimension, (ii) continuous Euclidean spaces of any dimension, (iii) bounded treewidth metrics, and (iv) planar metrics.*

We remark that our framework also extends to the setting when f is *asymmetric* norm. Note that asymmetric norms can potentially make the problem substantially harder. For example, a poly-time $\mathcal{O}(1)$-approximation algorithm exists for symmetric norms [34] but the asymmetric norm makes it $\Omega(\log n/\log\log n)$-hard to approximate even for the special case of ROBUST k-MEDIAN on the line metrics [24].

By *continuous* Euclidean space, we refer to the setting where any point of the space can be chosen as a center. This is in contrast to a *discrete* Euclidean space, where we restrict the centers to being selected from a specific finite subset of the points. Observe that for a fixed d, discrete Euclidean problems in $\mathbb{R}^d$ have a bounded doubling dimension, hence covered by our framework. Furthermore, it is not a shortcoming of our result that it does not cover discrete Euclidean spaces of high dimensions: we prove in Publication II that in the discrete case, k-CENTER is W[1]-hard to approximate within a factor of $\sqrt{3/2}-o(1)$.

Theorem 1.1.2 (Hardness of Discrete Euclidean)**.** *For any constant positive integer q, it is* **W[1]***-hard to approximate the discrete k-*CENTER *problem in $\mathbb{R}^{O(\log n)}$ under the ℓ_q metrics to within a factor of $(3/2-\eta)^{1/q}$ for any positive constant $\eta > 0$.*

Our main contributions in deriving the results of Theorem 1.1.1 have two parts: (i) a new concept of metric dimension and (ii) our main technical result showing EPASes for all the aforementioned clustering problems.

Unifying Metric Spaces via Scatter Dimension. Our key conceptual contribution is a new notion of bounded metric space dimension that relaxes the standard requirement of bounded doubling dimension so that the metric spaces mentioned in Theorem 1.1.1 all "live" in a finite dimension.

ϵ-scatter dimension. Given metric space $M = (P,F,\delta)$, where δ is a metric over $P \cup F$, the sequence sequence $(x_1,p_1)\ldots,(x_\ell,p_\ell)$ of center-point pairs $x_i \in F$, $p_i \in P$, $i \in [\ell]$ is said to be an *ϵ-scattering* if $\delta(x_i,p_j) \leq 1$ for all $1 \leq j < i \leq \ell$ and $\delta(x_i,p_i) > 1+\epsilon$ for all $i \in [\ell]$. The *ϵ-scatter dimension of M* is the maximum length of an ϵ-scattering in it.

A natural interpretation of such a sequence is a game between two players: the *center player* who tries to claim she can cover all the points with a unit ball and the *point player* who presents a counterexample. In the first round, the center player picks a center $x_1 \in F$, and the point player *refutes* the claim by presenting a point $p_1 \in P$ that is at least a factor $1+\epsilon$ away from the (closed) unit ball around x_1, that is, $p_1 \notin \mathsf{ball}(x_1, 1+\epsilon)$. The game continues this way: in the i-th round, the center player presents x_i such that $\{p_1,\ldots,p_{i-1}\} \subseteq \mathsf{ball}(x_i,1)$, and the point player gives $p_i \notin \mathsf{ball}(x_i,1+\epsilon)$. Both players are interested in prolonging the game as much as possible. The ϵ-scatter dimension is the length of the longest possible game. Another interpretation is to view such a sequence as a pair of ϵ-packings[2] that are required to be sufficiently distanced: It is easy to verify (simply using triangle inequalities) that $P^* = \{p_1,p_2,\ldots,p_{\ell-1}\}$ and $F^* = \{x_2,\ldots,x_\ell\}$ are ϵ-packings of the unit (closed) balls around x_ℓ and p_1, respectively. This view immediately implies that ϵ-scatter dimension is bounded in a bounded doubling metric.

Theorem 1.1.3. *For $\epsilon \in (0,1)$, any metric of doubling dimension d has ϵ-scatter dimension $(1/\epsilon)^{\mathcal{O}(d)}$.*

Next, we study the ϵ-scatter dimension of bounded treewidth graphs and show the following result using a delicate combinatorial argument.

Theorem 1.1.4. *For $\epsilon \in (0,1)$, the ϵ-scatter dimension is $\exp\left((1/\epsilon)^{\mathcal{O}(\mathsf{tw})}\right)$ for treewidth-tw graphs.*

Using the embedding result of [67] for planar graphs, the above result immediately implies the following.

Theorem 1.1.5. *For $\epsilon \in (0,1)$, the ϵ-scatter dimension is $\exp(\exp(\mathrm{poly}(1/\epsilon)))$ for planar graphs.*

Unfortunately, the bounded dimensionality does not hold in the high-dimensional (continuous) Euclidean metric[3]. To handle the high-dimensional continuous Euclidean setting, we present a stronger version of ϵ-scatter dimension that we call *algorithmic ϵ-scatter dimension*. The setting of

[2] For a metric space $M = (X,\delta)$ and a subset $Y \subseteq X$, we call $Y' \subseteq Y$ an ϵ-packing of Y if for all $y_1, y_2 \in Y$ we have that $\delta(y_1,y_2) > \epsilon$.

[3] To see this, consider the sequence $(x_1,p_1)\ldots,(x_{d-1},p_{d-1})$ where, for each $i \in [d-1]$, the point $x_i \in \mathbb{R}^d$ has i-th coordinate $1/\sqrt{2}$ and all other coordinates are zero. Define points $p_i = -x_i$ for all $i \in [d-1]$. It is easy to check that this sequence is a $(\sqrt{2}-1)$-scattering. This example implies that the ϵ-scatter dimension of continuous Euclidean metrics $\mathbb{R}^d$ can be at least $d-1$ (unbounded in ϵ).

the game is the same except that the center player would *optimize* to end the game early, while the point player would be interested in prolonging the game indefinitely. This means they play against each other. A *centering strategy* is a function $\sigma : 2^P \to F$ that specifies how the center $x_i = \sigma(\{p_1, \ldots, p_{i-1}\})$ would be chosen by the center player, given the points $p_1, \ldots, p_{i-1}$ played in the preceding rounds. The (σ, ϵ)-scatter dimension is the maximum number of rounds when the center player always plays strategy σ, and the algorithmic ϵ-scatter dimension is the minimum (σ, ϵ)-scatter dimension over all strategies σ. We remark that our actual definition is more involved, as it considers a weighted version of the game.

Theorem 1.1.6 (Bounding Algorithmic Scatter Dimension)**.** *The continuous Euclidean space (P, F, δ), that is, $P \subsetneq \mathbb{R}^d$ finite, and $F = \mathbb{R}^d$, has algorithmic ϵ-scatter dimension $\mathcal{O}(1/\epsilon^4 \log 1/\epsilon)$.*

EPAS for General Norm Clustering: Bypassing Coresets. Now we are ready to explain our main technical result that would allow us to obtain EPAS for all metrics having bounded ϵ-scatter dimension.

A generic tool whose existence immediately implies an EPAS is an ϵ-coreset —a "compression" of an input instance (P, F, δ) into a much smaller instance so that the cost of any solution is preserved within a factor of $(1 \pm \epsilon)$. The existence of an ϵ-coreset of size depending only on ϵ and k would immediately imply an EPAS (but not vice versa): First, use the ϵ-coreset to compress the instance (P, F, δ) to (P', F', δ') where $|P'| \le \gamma(\epsilon, k)$ and $\gamma(\cdot)$ is the function that upper bounds the size of the coreset P' of P. Then enumerate all possible partitioning of P' into k sets $P'_1, \ldots, P'_k$ (there are at most $k^{\gamma(\epsilon,k)}$ such partitions). For each set $i \in [k]$, compute the optimal center for P'_i. We choose the partition that has the lowest total cost.

This generic method, unfortunately, faces a serious information-theoretic limitation, that is, even for k-CENTER, ϵ-coresets of desirable sizes do not exist in high-dimensional Euclidean spaces [64]. Such lower bounds imply that one cannot hope to prove our (unified) results via the coreset route: While coresets are known for (k, z)-CLUSTERING for constant z [58]—allowing to handle k-MEANS and k-MEDIAN in a uniform fashion—it is impossible to extend this approach to k-CENTER. For more complex clustering objectives, such EPASes were in fact not known even for low dimensions. For example, the coreset of Braverman et al. [29] for ORDERED k-MEDIAN in $\mathbb{R}^d$ has size $\mathcal{O}_{\epsilon,d}(k^2 \log^2 n)$ and therefore does not give an EPAS even in low dimension.

Badoiu, Har-Peled, and Indyk [14] presented an EPAS for k-CENTER in high-dimensional Euclidean spaces (bypassing coresets in the above sense). Therefore, an obvious open question is whether their techniques can be extended to give an EPAS for any other clustering objective. Unfortunately, this is not even known for simple objectives such as PRIORITY k-CENTER. In fact, even the known EPASes for k-MEANS [90] and k-CENTER [14] are conceptually very different; to our knowledge, no approximation schemes handle k-MEANS and k-CENTER in a modular way. Our main technical result is presented in the following theorem. We remark that our techniques do not rely on any coreset constructions (thus bypassing the coreset lower bounds for k-CENTER).

Theorem 1.1.7. *Let $\mathcal{M}$ be a class of metric spaces that is closed under scaling distances by a positive constant. There is a randomized algorithm that computes for any* NORM k-CLUSTERING *instance $\mathcal{I} = (M, f, k)$ with metric $M = (P, F, \delta) \in \mathcal{M}$ and any $\epsilon \in (0,1)$ with high probability a $(1+\epsilon)$-approximate solution if the following two conditions are met.*

1. *There is an efficient algorithm evaluating for any distance vector $\boldsymbol{x} \in \mathbb{R}^P_{\geq 0}$ the objective $f(\boldsymbol{x})$ in time $T(f)$.*
2. *There exists a function $\lambda\colon \mathbb{R}_+ \to \mathbb{R}_+$, such that for all $\epsilon > 0$, the algorithmic ϵ-scatter dimension of $\mathcal{M}$ is at most $\lambda(\epsilon)$.*

The running time of the algorithm is $\exp\left(\widetilde{\mathcal{O}}\left(\frac{k\lambda(\epsilon/10)}{\epsilon}\right)\right) \cdot \mathsf{poly}(|M|) \cdot T(f)$.

Note that the complexity of computing f appears only as a linear factor in the running time. For instance, for SOCIALLY FAIR k-MEDIAN, the number m of groups affects only the computational cost of f, and therefore the running time is polynomial in m. Additionally, our algorithm is clean, simple, and entirely oblivious to both the objective and the structure of the input metric.

1.1.2 Robust Clustering in Discrete Geometric Spaces

Unfortunately, for discrete Euclidean space $\mathbb{R}^d$, the EPAS of Theorem 1.1.1 runs in time that is double exponential in the dimension d. This follows due to Theorem 1.1.3 which bounds ϵ-scatter dimension of discrete $\mathbb{R}^d$ by $(1/\epsilon)^{\mathcal{O}(d)}$, and hence Theorem 1.1.7 yields an EPAS for NORM k-CLUSTERING with running time $\exp\left(\widetilde{\mathcal{O}}(k/\epsilon^{\mathcal{O}(d)})\right) \cdot \mathsf{poly}(n) \cdot T(f)$. Specifically, for ROBUST (k,z)-CLUSTERING (or equivalently SOCIALLY FAIR (k,z)-CLUSTERING), which is a special case of NORM k-CLUSTERING, this result yields a running time of $\exp\left(\widetilde{\mathcal{O}}(k/\epsilon^{\mathcal{O}(d)})\right) \cdot \mathsf{poly}(n)$ in the discrete setting. The topic of Publication II is to obtain a more fine-grained understanding of the computational complexity of ROBUST (k,z)-CLUSTERING. Let us first define the problem of our interest.

Definition 1.1.1 (ROBUST (k,z)-CLUSTERING)**.** *Given a (clustering) metric space (P, F, δ) with $n = |P|$, positive integer k, a weight function $w\colon P \to \mathbb{R}_{\geq 0}$, and m groups $S_1, \ldots, S_m$ such that $S_i \subseteq P$ and $P = \cup_{i \in [m]} S_i$, find $X \subseteq F, |X| = k$ that minimizes* $\max_{i \in [m]} \sum_{p \in S_i} w(p)\delta(p, X)^z$.

Note that, in addition to generalizing k-MEDIAN and k-MEANS, the ROBUST (k,z)-CLUSTERING problem encapsulates k-CENTER when each group contains a distinct point.

The motivation for this problem originates in the domain of robust optimization wherein the task is to optimize a certain objective under the uncertainty of input data. In most practical applications, it is observed that the input data is rarely of high precision but often noisy, vulnerable to manipulation, or prone to errors. Low-precision data poses a serious algorithmic challenge; even slight perturbations in the input data can cause significant changes in the output of an algorithm, which can hinder its applicability in various contexts. Thus, in the definition of ROBUST (k,z)-CLUSTERING above, we are given m point sets (also called *groups* or *scenarios*) that represent the uncertainty in the actual point set, and our goal is to obtain a k-clustering of these points so as to minimize the (k,z)-CLUSTERING clustering cost. A special case of this problem when $z = 1$, known as ROBUST k-MEDIAN, was introduced by [8].

On the other hand, this problem also has connections to *algorithmic fairness* – here this problem, recently introduced independently by Abbasi et al. [4] and Ghadiri et al. [70], is known as SOCIALLY FAIR (k,z)-CLUSTERING. In this setting, the groups represent the demographic profiles of people (points), and our aim is to find a solution that is not substantially worse for any of the groups.

While k-MEANS, k-MEDIAN, and k-CENTER admit constant-factor approximations, it is not very surprising that ROBUST (k,z)-CLUSTERING is harder due to its generality: Makarychev and Vakilian [103] design a polynomial-time $\mathcal{O}(\log m/\log\log m)$-approximation algorithm, which is tight under a standard complexity assumption [24]. As this precludes the existence of efficient constant-factor approximation algorithms, recent works have focused on designing constant-factor parameterized approximation algorithms. Along these lines, an FPT time $(3^z+\epsilon)$-approximation algorithm has been proposed and shown to be tight under Gap-ETH [72]. When allowing a parameterization on the number of groups m (instead of k), Ghadiri et al. designed a $(5+2\sqrt{6}+\epsilon)^2$-approximation algorithm in $n^{\mathcal{O}(m^2)}$ time [71].

In Publication II, we address two main research questions related to ROBUST (k,z)-CLUSTERING in the discrete Euclidean $\mathbb{R}^d$. First, as mentioned before, the question is whether the doubly exponential dependency of d in the running time of EPAS of Publication I is necessary. We answer this question by designing a faster EPAS algorithm for doubling metrics, which generalizes $\mathbb{R}^d$.

Theorem 1.1.8 (EPAS for Doubling Metric of sub-logarithmic dimension)**.** *There is an algorithm that, given an instance of* ROBUST (k,z)-CLUSTERING *in a metric of doubling dimension d, computes a* $(1+\epsilon)$*-approximate solution for the instance, for every* $\epsilon > 0$*, in time* $f(k,d,\epsilon,z)\mathsf{poly}(m,n)$*, where* $f(k,d,\epsilon,z) = \left(\left(\frac{2^z}{\epsilon}\right)^d k\log k\right)^{\mathcal{O}(k)}$.

Note that the above result implies EPAS for the doubling metric of the sub-logarithmic dimension. Further, this running time is tight with respect to d, since in Theorem 1.1.2 we showed that it is **W[1]**-hard to approximate k-CENTER better than $\sqrt{3/2}$.

The second question we address is the parameterized approximablity of ROBUST (k,z)-CLUSTERING in high-dimensional discrete Euclidean space. While the best known lower bounds for parameterized approximation algorithms for ROBUST (k,z)-CLUSTERING, due to Goyal and Jaiswal [72], is the factor $3^z - o(1)$ in general discrete metrics, it does not automatically hold true for the important case of the (discrete) high-dimensional Euclidean space. In this direction, we show that the structures of Euclidean spaces can be leveraged to obtain better results in high dimensions.

Theorem 1.1.9 (High Dimensional Setting)**.** *There exists a universal constant $\eta_0 > 0.0006$ such that for any constant positive integer z, there is a factor $3^z(1-\eta_0)$ parameterized approximation algorithm for* ROBUST (k,z)-CLUSTERING *in discrete $\mathbb{R}^d$ that runs in time $2^{\mathcal{O}(k\log k)}\mathsf{poly}(m,n,d)$.*

1.1.3 Diversity-aware Clustering: Ensuring Fairness in Center Selection

In the previous section, we saw parameterized approximation algorithms for ROBUST (k,z)-CLUSTERING, which is also known as SOCIALLY FAIR (k,z)-CLUSTERING in the algorithmic fairness community. *Algorithmic fairness*, particularly fair clustering [1], has received significant attention from the research community in the past half-decade. The high-level goal here is to guarantee certain notion of fairness in the chosen solution. In clustering, there are at least 10 notions of fairness (see [1] for a survey and references therein). For example, SOCIALLY FAIR (k,z)-CLUSTERING ensures fairness by obtaining a solution that is simultaneously good for all the groups. In Publication III, we consider a notion of fairness for the centers in the solution. This problem, introduced by [122], is called *Diversity-aware Clustering*, and the goal is choose k centers to minimize certain objective while avoiding under-representation and over-representation of communities defined over the facility set F. This problem naturally arises, among many others, in the context of committee selection, where the goal is to form a committee that is not only easily accessible (in terms of distances) but is also as diverse as possible. Now, let us define a special case of this problem, which will be of interest in this section.

Definition 1.1.2 (Diversity-aware k-MEDIAN (DIV-k-MED))**.** *Given a metric space (V,δ), a set $C \subseteq V$ of clients, a set $F \subseteq V$ of facilities, an integer $0 \le k \le |F|$, and a collection, called groups, $\mathcal{G} = \{G_1,\cdots,G_m\}, G_i \subseteq F$ with integers $0 \le r_i \le R_i \le k$, the problem is find $S \subseteq F$, $|S| = k$ satisfying $r_i \le |S \cap G_i| \le R_i$ for all $i \in [m]$ such that the clustering cost of S, $\mathsf{cost}(S) = \sum_{c\in C} \delta(c,S)$ is minimized.*

As this problem generalizes the classical k-MEDIAN, it inherits all its hardness. In fact, in Publication III, we show that this problem is much harder – it is even hard to approximate when parameterized by k. Our main result is the following parameterized approximation algorithm for the problem when parameterized by (k,m) which is also tight under a plausible assumption.

Theorem 1.1.10. *There exists a randomized* $(1+2/e+\epsilon)$*-approximation algorithm for* DIV-k-MED*, for* $\epsilon > 0$*, running in time* $f(k,m,\epsilon)\cdot \mathrm{poly}(|U|)$*, where* $f(k,m,\epsilon) = \mathcal{O}\left(\left(\frac{2^m k^3 \log^2 k}{\epsilon^2 \log(1+\epsilon)}\right)^k\right)$*. Further, assuming Gap-ETH, the approximation ratio is tight for any parameterized algorithm parameterized by* (k,m)*.*

A surprising element of the above theorem is that, when allowed to use additional parameter m, it matches the approximation factor guarantee of k-MEDIAN, which is a considerably simpler version of our problem.

1.2 Graph Packing

In the second part of this thesis, we switch gears and design lower bounds for graph packing problems. Graph packing problems are among the central problems in combinatorial optimization and have been a subject of extensive research for more than half a century now. Prominent problems under this umbrella include *Maximum Independent Set in graphs, Set Packing (or Maximum Hypergraph Matching), Knapsack, Interval Scheduling,* and many more. In this thesis, we consider two of such problems – *Set Packing* and *Maximum Independent Set in k-claw-free graphs* – and understand their computational complexity under various computational models.

1.2.1 Set Packing in a Compact Universe

The SET PACKING problem is given a set system $(\mathcal{U},\mathcal{S})$ and a positive integer r to find r sets in $\mathcal{S}$ that are pairwise disjoint. That is, we want to find a packing of r sets in $\mathcal{S}$. This problem is one of the fundamental problems in combinatorial optimization with numerous applications [127]. While this problem captures many classical combinatorial problems such as maximum independent set (or maximum clique) [9], k-dimensional matching, and other graph packing problems [44, 80], this generalization also makes the problem intractable in several regimes. For instance, the trivial algorithm that enumerates all r-sized subsets of $\mathcal{S}$ running in time $\mathcal{O}^*(|\mathcal{S}|^r)$ is the (asymptotically) fastest algorithm to hope for, unless *Exponential Time Hypothesis (ETH)* fails. Similarly, even finding non-trivial approximations to the problem is notoriously hard [81, 78]. To combat these negative results, several restrictions on the problem have been studied. One regime where SET PACKING has been studied extensively is the parameterized complexity regime. Here, the problem is known as PARAMETERIZED SET PACKING (PSP) with parameter r. Unfortunately, PSP remains intractable and is, in fact, **W[1]**-complete [63]. Even from the parameterized approximations perspective, the problem does not admit any non-trivial factor [36], assuming *Gap-ETH*, a stronger version of ETH. Very recently, assuming a weaker hypothesis of $\mathsf{FPT} \neq \mathbf{W[1]}$, [114] ruled out an FPT algorithm that can find a packing of size $r/r^{1/H(r)}$ for any increasing function $H(\cdot)$, when given a promise that there is an r-packing in the instance. Thus, SET PACKING is a notoriously hard problem in almost all computational regimes, even with restrictions.

In Publication IV, we consider yet another restriction of SET PACKING, which we call the *compact* regime. Here, we study the computational complexity of PSP on compact instances – instances that have a small universe. More precisely, in the COMPACT PSP problem, we have that $|\mathscr{U}| = f(r)\cdot\Theta(\mathsf{poly}(\log|\mathscr{S}|))$, for some function $f(r) \geq r$. Apart from algorithmic motivation, compact instances have recently been used as an intermediate step to show FPT-inapproximibility of (non-compact) classical problems (see, e.g., [25, 99] where the compact instances were used in proving FPT-inapproximability of the k-EvenSet and Dominating Set). We hope that studying COMPACT PSP would lead to some ideas that would be useful in proving tight FPT inapproximability of PSP (that is, to weaken the Gap-ETH assumption used in [36]). We would like to remark on the choice of defining compact instance on a small universe size and not on a small $|\mathscr{S}|$. Note that, when $|\mathscr{S}| = f(r)\cdot\Theta(\mathsf{poly}(\log|\mathscr{U}|))$, the enumeration algorithm running in time $\mathscr{O}^*(|\mathscr{S}|^r)$ already yields an FPT algorithm [33].

Our main result is the following dichotomy of PARAMETERIZED SET PACKING.

Theorem 1.2.1 (Dichotomy)**.** *The following dichotomy holds for* PSP.

- *If* $|\mathscr{U}| = f(r)\cdot o(\log|\mathscr{S}|)$*, for any* f*, then* PSP *is in FPT.*
- PSP *remains* **W[1]***-hard even when* $|\mathscr{U}| = r\cdot\Theta(\log|\mathscr{S}|)$.

The algorithmic result follows from well-known dynamic programming based algorithms [26, 60] that run in time $O^*(2^{|\mathscr{U}|})$ which is FPT time when $|\mathscr{U}| = f(r)o(\log|\mathscr{S}|)$. The main contribution of our work is the **W[1]**-hardness of PSP even when $|\mathscr{U}| = r\cdot\Theta(\log|\mathscr{S}|)$. To this end, we show an FPT-reduction from SUBGRAPH ISOMORPHISM (SGI) to COMPACT PSP. The hardness result follows since SGI is **W[1]**-hard. In fact, our hardness result can be strengthened by assuming the Exponential Time Hypothesis (ETH) [60] to obtain the following result.

Theorem 1.2.2. COMPACT PSP *requires time* $|\mathscr{S}|^{\Omega(r/\log r)}$ *even when* $|\mathscr{U}| = r\cdot\Theta(\log|\mathscr{S}|)$*, unless ETH fails.*

An interesting consequence of the dichotomy theorem coupled with Theorem 6.3.2 is the fact that, as soon as instances get asymptotically smaller, not only do we beat the enumerative algorithm, but we actually obtain an FPT algorithm.

Further, we extend our framework to obtain lower bounds for compact versions of other combinatorial problems.

1.2.2 Maximum Independent Set in k-claw-free Graphs using Convex Relaxations

In Publication V, we consider the computational complexity of the structural properties of connected k-claw-free graphs, which are generalization of claw-free graphs. A graph is said to be claw-free if it does not contain induced $K_{1,3}$. More generally, a graph is k-claw-free, for $k \geq 3$, if it does not contain induced $K_{1,k}$. Further, for a graph G, $\alpha(G)$, $\omega(G)$, and $\chi(G)$ denote the size of maximum independent set in G, the size of maximum clique in G, and the chromatic number of G, respectively. Our work focuses on an extremal question in k-claw-free graphs and its connection to the power of convex programs in estimating the *Maximum Weight Independent Set (MWIS)* in such graphs. The study of such a relation originated already around 50 years ago, when Lovász defined the notion of perfect graphs based on graph extremal properties and showed connections to (exact) semi-definite programming formulations for optimization problems [74, 102]. Such connections are known to be "approximation preserving" so they imply a connection between standard Ramsey-type theorems and χ-boundedness in approximating the cardinality (resp. the weight) of the maximum independent set. Several approximation algorithms for geometric intersection graphs have been successfully derived in this framework [96, 35, 39, 40].

Most prior works that extend perfect graphs rely on the notion of the clique-constrained stable set polytope (QSTAB)—a convex relaxation that can be optimized via semi-definite programs.[4] The power of QSTAB is captured precisely by standard extremal bounds. For example, the χ-boundedness $\chi(G) \leq \gamma\omega(G)$ in a "natural" graph class is (roughly) equivalent to QSTAB providing a γ-estimate on the weight of the size of the maximum independent set in the same graph class [38]. Despite successful cases of the extremal approach, we observe that QSTAB fails unexpectedly in graph classes such as k-claw-free graphs: For any $k \geq 3$, a simple greedy algorithm immediately gives a factor $(k-1)$ approximation for MWIS, while QSTAB (and other known convex relaxations) is unable to give a $f(k)$ approximation for any function f.

Our work is an attempt to better understand the power of convex relaxations for approximating MWIS in k-claw-free graphs. MWIS on k-claw-free graphs contains many well-known (open) problems as special

[4]The polytope is defined as $\mathsf{QSTAB}(G) = \{x \in [0,1]^{|V(G)|} : \sum_{i \in Q} x_i \leq 1 (\forall \text{clique } Q)\}$. Optimizing this itself is NP-hard, but we can optimize an SDP whose solution is feasible for QSTAB.

cases, such as set packing [59, 43, 82, 83] and independent set in sparse graphs [18, 11], for which QSTAB has been shown to perform relatively well in terms of approximating the problems. Somewhat surprisingly, the study of convex relaxations for MWIS on k-claw-free graphs has been absent from the literature.

The motivation for this study comes from the paper of Chudnovsky and Seymour [49], where they studied the conditional variant of χ-boundedness and showed that any claw-free graph G with an independent set of size three must have $\chi(G) \le 2\omega(G)$. This result, apart from being mathematically interesting, actually yields an efficient algorithm to estimate the size of the maximum independent set in claw-free graphs within factor 2. The aforementioned 2-approximation algorithm is the sum-of-squares (SoS) hierarchy algorithm with three rounds. Thus, an interesting question is whether similar conditional bounds on $\chi(G)$ exist when G is a k-claw-free graph. An affirmative answer to this question yields a SoS approximation algorithm for the size of the maximum independent set in k-claw-free graphs. As mentioned earlier, the simple greedy algorithm of picking an arbitrary maximal independent set achieves factor $(k-1)$-approximation. Hence, it is an interesting algorithmic question if SoS can approximately estimate the size of the maximum independent set in k-claw-free graphs in constant rounds even within a factor of $f(k)$ for some function f. In our Publication V, we formally initiate the study of connections between the χ-boundedness of k-claw-free graphs and the integrality gap of SoS.

Further, we negatively answer the question of whether $\chi(G) \le f(k)\omega(G)$ for the k-claw-free graph G, even when $\alpha(G) = \tilde{\Omega}(|V(G)|)$, refuting the possibility of generalizing the theorem of Chudnovsky and Seymour, even for $k = 4$.

Theorem 1.2.3. *For $k \ge 4$, there exists n_0 depending on k such that for infinitely many $n \ge n_0$, there exists a connected k-claw-free graph G on n vertices with $\alpha(G) = \Omega\left(\frac{n}{\log n}\right)$ and $\chi(G) \ge f(k)\left(\frac{\omega(G)}{\log \omega(G)}\right)^{k/2}$, for some $f(k)$.*

Finally, regarding the power of SoS, we show evidence that it may perform much worse than the simple greedy algorithm mentioned above. In this direction, we use the counter example of the previous question to construct a bad example for *Sherali–Adams* hierarchy, which, although weaker than SoS, is known to provide tight results for special cases of k-claw-free graphs. The notations O_k, Θ_k, Ω_k in the following theorem hide multiplicative functions in k.

Theorem 1.2.4 (Integrality gap of Sherali-Adams on QSTAB)**.** *Let $0 < \epsilon \le 1/3$. For $k \ge 4$, for infinitely many n, there exists a connected k-claw-free graph G_n on n vertices with $\alpha(G_n) = \Theta_k(n^\epsilon)$ such that the integrality gap of Sherali–Adams hierarchy on $\mathsf{QSTAB}(G_n)$ is $\Omega_k(n^\epsilon)$, even after $\Omega_k(n^{1-2\epsilon})$ rounds.*

Discussion of previous work on convex programs. Let us compare our result of Theorem 1.2.4 with the bounds of Chan and Lau [41]. In particular, [41] considers k-SET PACKING, a special case of maximum independent set in $(k+1)$-claw-free graph, where we are given a k-uniform hypergraph H on n vertices and we are asked to find a maximum matching in H. In their work, [41] show that Sherali–Adams on the standard LP for k-SET PACKING has an integrality gap of at least $k-2$, even after $\Omega_k(n)$ rounds. Additionally, they show that, for constant k, QSTAB for k-SET PACKING can be captured by a polynomial size LP and has an integrality gap of at most $(k+1)/2$. In contrast, our result of Theorem 1.2.4 is for a more general problem of maximum independent set in k-claw-free graphs and yields an integrality gap of $\Omega_k(n^\epsilon)$, which is a function of n, for Sherali–Adams on QSTAB with rounds $\Omega_k(n^{1-2\epsilon})$, which is a stronger program than that of [41]. However, we remark that the results of Theorem 1.2.4 can be adjusted to yield an integrality gap of $g(k)$ for any function g for Sherali–Adams on QSTAB with $\Omega_k(n)$ number of rounds, which is linear in n. Thus, our results yield a larger integrality gap for a stronger program for maximum independent set in k-claw-free graphs compared to that of [41].

1.3 Organization

This thesis is divided into two parts. In Part I, we design parameterized approximation algorithms for several clustering problems. Specifically, in Chapter 3 we present an overview of the EPAS framework of Publication I for NORM k-CLUSTERING. In Chapters 4 and 5, we present an overview of the parameterized approximation algorithms of Publication II for ROBUST (k,z)-CLUSTERING and that of Publication III for Diversity-aware k-MEDIAN, respectively. In Part II, we design lower bounds for graph packing problems, starting with Chapter 6, where we present the parameterized intractability result of Publication IV for COMPACT PSP, and finally, in Chapter 7, we present the results of Publication V for MWIS in k-claw-free graphs. We present preliminaries in Chapter 2. The original publications are appended at the end of this thesis.

Each chapter starts with an introduction and motivation addressing the problems of interest, accompanied by a discussion of related work. Subsequently, the chapter offers a comprehensive overview of our results and showcases specifically selected results to underscore their novelty. The chapter concludes by addressing open problems for further exploration. The proofs of other intricate results and associated mathematical concepts are deferred to the respective publications, which are appended at the end of the thesis. It is noteworthy that, given the article-based format of this thesis, the original publications appended are integral components of the thesis.

2. Preliminaries

2.1 Clustering

This section begins by introducing fundamental terminology relevant to clustering and subsequently delves into basic clustering problems.

2.1.1 Terminology

Classes of Metric Clustering Spaces. A *metric clustering space* (or metric space for brevity) is a triple $M = (P, F, \delta)$ where P is a finite set of n *data points*, F is a (possibly infinite) set of potential locations of cluster centers, and δ is a metric on $P \cup F$. Sets P and F are not necessarily disjoint. (For example, it is natural for clustering problems to have $P = F$ or $P \subseteq F$.) By $|M|$, we denote the space needed to represent the metric space M in the memory. If M is finite, then $|M|$ is polynomial in $|F|$, $|P|$, and the space needed for storing a point and a center, respectively. If F is infinite (for example, in the continuous Euclidean setting, $F = \mathbb{R}^d$), $|M|$ is polynomial in $|P|$ and the space of storing a point. A *class* $\mathscr{M}$ of metric spaces is an (infinite) set of metric spaces.

Let $M = (P, F, \delta)$ be a metric clustering space. For $X \subseteq F$ and $p \in P$, let $\delta(p, X) = \min_{x \in X} \delta(p, x)$.

2.1.2 Basic Clustering Problems

Let $M = (P, F, \delta)$ be a metric clustering space. Further, every point $p \in P$ has a weight $w(p) \geq 0$. Let $k \geq 1$ and $z \geq 1$ be positive integers. In each of the following problems, we are asked to find a k-element subset $X \subseteq F$ that minimizes

k-MEDIAN: $\sum_{p \in P} w(p)\delta(p, X)$

k-MEANS: $\sum_{p \in P} w(p)\delta(p, X)^2$

k-CENTER: $\max_{p\in P} w(p)\delta(p,X)$

(k,z)-CLUSTERING: $\sum_{p\in P} w(p)\delta(p,X)^z$.

2.2 Optimization Problems on Graphs

We define relevant graph problems in this section. Let $G=(V,E)$ be a graph.

- An *independent set* $I \subseteq V$ in G consists of vertices that have no edges between them. $\alpha(G)$ represents the size of the largest independent set in G. When G is a weighted graph, $\alpha(G)$ represents the maximum weight of an independent set in G.
- A *clique* $Q \subseteq V$ in G consists of vertices where every pair shares an edge. $\omega(G)$ represents the size of the largest clique in G. In weighted graphs, $\omega(G)$ represents the maximum weight of a clique in G.
- The chromatic number of G, denoted as $\chi(G)$, is the minimum number of colors required to color the vertices of G so that no two vertices sharing an edge receive the same color.

It is important to note that computing $\alpha(G)$, $\omega(G)$, and $\chi(G)$, even for an unweighted graphs, is **NP**-hard.

2.3 Algorithmic Regimes for NP-hard Problems

2.3.1 Approximation Algorithms

As mentioned in the introduction, many fundamental problems in combinatorial optimization are **NP**-hard, and one way to move further is to relax our ambition and look for *Approximation Algorithms*. These algorithms run in polynomial time with a guarantee that their solution has a *cost* that is within some factor of the optimal cost. For example, a constant-factor approximation algorithm produces a solution whose cost is within some constant factor of the optimal cost. Particularly for minimization problems like computing $\chi(G)$, a factor c-approximation algorithm, for $c>1$, will output a solution in polynomial time, which, in the worst case, is c times the optimal cost. Conversely, for maximization problems like computing $\alpha(G)$ and $\omega(G)$, a factor c-approximation algorithm, for $c>1$, will provide a solution in polynomial time that is at least $1/c$ of the optimal in the worst case. This guarantee of how close the solution is to the optimal cost is referred to as the algorithm's *approximation factor*. Within the realm of approximation algorithms, there is a special class of algorithms known as *Efficient Polynomial Time Approximation Schemes* (EPTAS). These schemes go a step further, providing a solution that is $(1+\epsilon)$-approximate for every $\epsilon > 0$ in time $f(\epsilon)\mathsf{poly}(n)$ for some function f, where n is the size of the input instance. Going further, there is a class called *Polynomial Time Approximation Schemes* (PTAS). PTAS also provides a solution that is $(1+\epsilon)$-approximate for every $\epsilon > 0$, but with a runtime of $f(\epsilon)n^{g(\epsilon)}$, for some functions f and g. [128, 126] are excellent references for approximation algorithms.

2.3.2 Parameterized Complexity

The parameterized complexity theory concerns computational aspects of languages (L, κ) over a fixed and finite alphabet Σ, where $L \subseteq \Sigma^*$ and $\kappa : \Sigma^* \to \mathbb{N}$, called the parameter, is a polynomial time computable function. Thus, a parameterized problem is a classical problem with a parameter κ. As an example, consider the following classical NP-complete problem.

CLIQUE	
Instance:	A graph G and $k \in \mathbb{N}$
Problem:	Decide if G has a clique of size k

Now consider a parameterized version of CLIQUE defined by $\kappa(G,k) := k$.

k-CLIQUE	
Instance:	A graph G and $k \in \mathbb{N}$
Parameter:	k
Problem:	Decide if G has a clique of size k

When the parameter κ represents the size of the solution, it is called a *natural* parameter. Next, we define when a parameterized problem is tractable in this regime.

Definition 2.3.1 (Fixed Parameter Tractable)**.** *A parameterized problem (L,κ) is called fixed parameter tractable if there is an algorithm A, a constant c, and a computable function $f : \mathbb{N} \to \mathbb{N}$ such that on all inputs $y = (x,k)$, A decides whether x is in L and runs in time at most $f(k) \cdot |x|^c$. Furthermore, algorithm A is called* parameterized algorithm *for the problem (L,κ).*

The complexity class FPT is the set of all fixed-parameter tractable problems. In this thesis, unless stated, we consider parameterized problems with natural parameters, i.e., κ represents the size of the solution. Once we define the class FPT, the next natural thing is to define *parameterized reduction* or *FPT-reduction* with the intention that such a reduction from parameterized problem Q to another parameterized problem Q' allows converting an FPT algorithm of Q' to an FPT algorithm of Q.

Definition 2.3.2 (FPT-reduction). *An FPT-reduction from $Q \subseteq \Sigma^* \times \mathbb{N}$ to $Q' \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm R mapping from $\Sigma^* \times \mathbb{N}$ to $\Sigma^* \times \mathbb{N}$ such that for all $y = (x,k), R(y) \in Q'$ if and only if $y \in Q$, and for some computable function f and a constant c, $R(y)$ runs in time $f(k) \cdot |x|^c$ and $R(y) = (x',k')$, where $k' \leq g(k)$ for some computable function g.*

Definition 2.3.3 (Class **W[1]**). *A parameterized problem (L,κ) is in class* **W[1]** *if there is an FPT reduction from (L,κ) to k-*CLIQUE*. Further, (L,κ) is* **W[1]***-hard if there is an FPT reduction from k-*CLIQUE *to (L,κ).*

We remark that the actual definition of class **W[1]** is quite technical; however, the above definition is equivalent to the actual definition.

An extensive treatment of the subject can be found in [60, 63, 66].

2.3.3 Parameterized Approximation Algorithms

Nevertheless, many important problems are individually quite hard for both approximation algorithms and parameterized algorithms. In response to this intractability, a promising approach emerges: the realm of *Parameterized Approximation Algorithms* which is also referred to as *FPT Approximation Algorithms*. The goal in this setting, as the name suggests, is to obtain improved factor approximation algorithms, even if they are allowed to run in superpolynomial time. Specifically, for a chosen parameter, denoted as κ, of the input, these algorithms run in time $f(\kappa)\mathsf{poly}(n)$, where f is a function, and n is the size of the input. Typical parameters for clustering problems include the solution size (k), treewidth (tw) of the underlying metric space, and dimension (d) of the data points, among others.

The rationale behind permitting superpolynomial running time is that, most often in practice, these parameters are significantly smaller compared to the input size, n. For instance, in the clustering domain, k is typically less than 10 in practice, while n could be on the order of millions. Consequently, these algorithms effectively behave as polynomial-time algorithms when the parameters are small, offering the potential for vastly improved approximation factors compared to standard approximation algorithms.

Within the realm of parameterized approximation algorithms, there is a subset known as *Efficient Parameterized Approximation Schemes* (EPAS). These schemes are designed to provide a $(1+\epsilon)$ approximate solution running in time $f(k,\epsilon)\mathsf{poly}(n)$ for every $\epsilon > 0$. It is worth noting that PTASes (Polynomial Time Approximation Schemes) are incomparable to EPASes. Furthermore, there is a more relaxed subset of algorithms known as *Parameterized Approximation Schemes* (PAS). These algorithms, like EPAS, provide a $(1+\epsilon)$ approximate solution, but run in time $f(k,\epsilon)n^{g(\epsilon)}$ for every $\epsilon > 0$. Figure 2.1 depicts the connections between various approximation schemes.

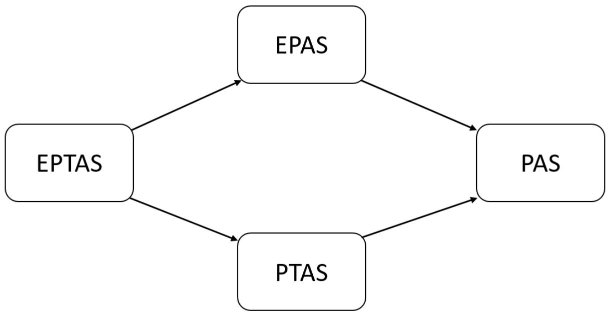

Figure 2.1. Various Approximation Schemes. Existence of the tail implies the existence of the head of the arrow. For example, a problem admitting EPTAS implies that the problem admits all other approximation schemes.

Part I

Algorithms for Clustering: Fairness, Robustness, and Beyond

3. Clustering with General Norm Objectives

Clustering is among the most fundamental computational problems that arise in optimization, machine learning, data mining, and computational geometry. It has received significant attention from the algorithmic community for over half a century. In the center based k-clustering problem, we are given n data points P in a metric space and are asked to partition P into k *clusters* along with a representative (*center*) for each cluster to minimize a certain clustering objective that is a function of distances between the points and their centers. Classical clustering problems include k-MEDIAN, k-MEANS, and k-CENTER, with objectives being the sum of distances, the sum of squared distances, and the maximum distance of a point to its center, respectively.

In this chapter and the following two chapters, we are interested in designing *parameterized approximation algorithms* for various clustering problems in various metric spaces. Particularly in this chapter, we design an EPAS framework for clustering objective that is a *monotone norm* of the distances in several metric spaces. This objective captures essentially all clustering objectives studied in the literature and also extends to new objectives that are interesting in their own right. We remark that EPASes have been prominent in the literature for many classical clustering problems [79, 105, 108, 62, 23, 30, 57].

3.1 Notation

Given a clustering space $M = (P, F, \delta)$ on $n = |P|$ data points, let $\boldsymbol{b} \in \mathbb{R}^{P}_{\geq 0}$ be an n-dimensional vector. We interpret $\boldsymbol{b}$ as assigning each point $p \in P$ a non-negative value denoted $b(p)$. That is, $\boldsymbol{b} = (b(p))_{p \in P}$. For example, given a subset $X \subseteq F$ of centers, we define the *distance vector* $\boldsymbol{\delta}(P, X) = (\delta(p, X))_{p \in P}$, where $\delta(p, X) = \min_{x \in X} \delta(p, x)$. Given any point $u \in P \cup F$ and a radius $r \in \mathbb{R}_{+}$, we denote by $\mathsf{ball}_{\delta}(u, r) = \{v \in P \cup F \mid \delta(u, v) \leq r\}$ the ball of radius r centered around u. We drop the subscript δ if the distance function is clear from the context.

If $B \subseteq P$ is a subset of points, then $\mathbf{1}_B \in \{0,1\}^P$ denotes the *characteristic vector of* B, that is, it assigns value 1 to any $p \in B$ and 0 to any $p \in P \setminus B$. If $p \in P$ and $\alpha \geq 0$, then we denote by $\mathbf{1}_{p,\alpha}$ the binary vector $\mathbf{1}_{\mathsf{ball}(p,\alpha)\cap P}$, where $\mathsf{ball}(p,\alpha) \cap P$ denotes the set of points of P in the $\mathsf{ball}(p,\alpha)$. Given a metric space $M = (X, \delta)$ and a subset $Y \subseteq X$, we call $Y' \subseteq Y$ an ϵ-packing of Y if for all $y_1, y_2 \in Y$ we have that $\delta(y_1, y_2) > \epsilon$.

3.2 NORM k-CLUSTERING

Definition 3.2.1 (Norm, Monotone norm)**.** *A norm is a function* $f : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, $n \in \mathbb{N}$ *that satisfies (i) for all* $\boldsymbol{x} \in \mathbb{R}^n$, $f(\boldsymbol{x}) = 0$ *if and only if* $\boldsymbol{x} = \mathbf{0}$, *(ii)* $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n : f(\boldsymbol{x}+\boldsymbol{y}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y})$, *and (iii)* $\forall \boldsymbol{x} \in \mathbb{R}^n, \lambda \in \mathbb{R}: f(\lambda \boldsymbol{x}) = |\lambda| f(\boldsymbol{x})$. *We say that* f *is monotone if* $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$ *whenever* $\boldsymbol{x} \leq \boldsymbol{y}$, *where the inequality is co-ordinate wise inequality.*

Definition 3.2.2 (NORM k-CLUSTERING)**.** *Given n data points P, candidate centers F, a metric space* $M = (P \cup F, \delta)$, *a positive integer k, and a monotone norm function* $f : \mathbb{R}^P \to \mathbb{R}_{\geq 0}$, *the goal of* NORM k-CLUSTERING *is to compute* $X \subseteq F, |X| = k$ *that minimizes* $f(\boldsymbol{\delta}(P, X))$.

While Chakrabarty and Swamy [34] further require that f be symmetric (we say that f is symmetric if $f(\boldsymbol{x}) = f(\boldsymbol{x}')$ whenever $\boldsymbol{x}'$ can be obtained by reordering coordinates of $\boldsymbol{x}$), our techniques in this chapter apply to all monotone norm functions. This family includes the following well-known clustering problems (see Figure 3.1 for an overview):

- **From k-MEANS, k-CENTER, and k-MEDIAN to (k,z)-CLUSTERING**: All basic clustering problems can be captured by the ℓ_z-norm when $z \in \{1, 2, \infty\}$. In fact, a (k,z)-clustering problem [84, 58, 56] (for constant positive integer z) uses the objective function $g(\boldsymbol{x}) = \sum_{p \in P} |x(p)|^z$. (This function itself is not a norm, but we can instead consider the ℓ_z-norm $f(\boldsymbol{x}) = g(\boldsymbol{x})^{1/z}$.)

- **WEIGHTED k-CENTER (or PRIORITY k-CENTER)**: The weighted version of k-CENTER [98, 15, 112] generalizes k-CENTER so that each data point $p \in P$ is associated with a positive weight (or priority) $w(p)$, and the objective is to minimize the (weighted) maximum distance to a center.[1] This problem can be modeled by the "weighted max" norm $f(\boldsymbol{x}) = \max_{p \in P} w(p) x(p)$. One can analogously define the weighted versions of k-MEDIAN and k-MEANS (see, for example, [51]). We remark that the underlying weighted norms are not symmetric.

- **ℓ-CENTRUM**: This problem (sometimes called k-FACILITY ℓ-CENTRUM) aims to minimize the sum of the connection costs among

[1] For convenience of presentation, the terminologies we use are somewhat different from the literature.

the ℓ "most expensive" points (that is, those that are furthest away from the open centers). The problem generalizes both k-CENTER ($\ell = 1$) and k-MEDIAN ($\ell = |P|$) problem [120]. (See the books [107, 92] for more details on ℓ-CENTRUM and the more general ORDERED k-MEDIAN discussed below.) This problem can be modeled by the top-ℓ norm $f(\boldsymbol{x}) = \sum_{j=1}^{\ell} x^{\downarrow}(j)$ where $\boldsymbol{x}^{\downarrow}$ denotes the reordering of vector $\boldsymbol{x}$ so that the entries appear non-increasingly. The top-ℓ norm is symmetric.

- **ORDERED k-MEDIAN**: This problem further generalizes ℓ-CENTRUM, allowing flexible penalties to be applied to data points that incur the highest connection costs. More formally, the objective is the *ordered weighted norm* $f(\boldsymbol{x}) = \boldsymbol{v}^{\intercal}\boldsymbol{x}^{\downarrow}$ where $\boldsymbol{v} \in \mathbb{R}^n_{\geq 0}$ is a non-increasing cost vector, that is, $v(1) \geq v(2) \geq \ldots \geq v(n)$. ℓ-CENTRUM corresponds to $\boldsymbol{v} = (1,\ldots,1,0,\ldots 0)$ where the first ℓ-entries of $\boldsymbol{v}$ are ones. This problem has already received attention for a few decades [32, 34, 29]. We remark that f here is a monotone and symmetric norm.

- **SOCIALLY FAIR k-MEDIAN (or ROBUST k-MEDIAN)**: In SOCIALLY FAIR k-MEDIAN, along with the set of points P, we are given m different (not necessarily disjoint) subgroups such that $P = \bigcup_{i\in[m]} P_i$. Our goal is to find a set X of centers that incurs fair costs for the groups by minimizing the maximum cost over all the groups. In other words,
$$\min_{\substack{X \subseteq F\\ |X|=k}} \max_{i\in[m]} \sum_{p\in P_i} \delta(p,X).$$
Due to distinct applications in at least two domains, this variant of clustering has recently been studied extensively: (i) in algorithmic fairness [5, 73, 103, 71] and (ii) in the robust optimization context, this problem is known as ROBUST k-MEDIAN, which intends to capture the applications when we are uncertain about the actual data scenarios (corresponding to the groups P_i) that may come up [8, 24, 20].

- **(z,q)-FAIR CLUSTERING:** Our problem also models a clustering problem called (z,q)-FAIR CLUSTERING[2] introduced by Chlamtáč et al. [47], which generalizes SOCIALLY FAIR k-MEDIAN. In particular, one can view the cost function f of SOCIALLY FAIR k-MEDIAN as a "two-level" aggregate cost: First, cost $\sum_{p\in P_i} \delta(p,X)$ incurred by group P_i, $i \in [m]$ can be viewed as weighted ℓ_1-norm $\boldsymbol{w}_i^{\intercal}\boldsymbol{x}$ where $\boldsymbol{w}_i = \mathbf{1}_{P_i} \in \{0,1\}^P$ denotes the characteristic vector of P_i. Second, these group costs are further aggregated through ℓ_∞, that is, $f(\boldsymbol{x}) = \max(\boldsymbol{w}_1^{\intercal}\boldsymbol{x}, \boldsymbol{w}_2^{\intercal}\boldsymbol{x}, \ldots, \boldsymbol{w}_m^{\intercal}\boldsymbol{x})$.

[2] Chlamtáč et al. [47] call the problem (p,q)-FAIR CLUSTERING. For the sake of consistency with the notation in the rest of the paper, we changed the naming slightly.

(z,q)-FAIR CLUSTERING allows arbitrary uses of the ℓ_z and ℓ_q norms to aggregate costs on two levels. The cost function is defined as $f(\boldsymbol{x}) = g(\boldsymbol{h}(\boldsymbol{x}))$ where g is any ℓ_q-norm function and $\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x}))$ where $h_i(\boldsymbol{x})$ is a weighted ℓ_z-norm, that is, $h_i(\boldsymbol{x}) = \left(\sum_{p \in P} w_i(p) x(p)^z\right)^{1/z}$ for arbitrary weight vectors $\boldsymbol{w}_i \in \mathbb{R}^P_{\geq 0}$, $i \in [m]$. It is easy to check that $f(\boldsymbol{x}) = g(\boldsymbol{h}(\boldsymbol{x}))$ is a monotone norm whenever g and $\{h_i\}$ are.

- **Beyond the Known Problems**: Our (asymmetric) norm formulation allows us to model more complex clustering objectives that might be useful in some application settings and, to our knowledge, have not yet been considered in the algorithms community. One such objective is the PRIORITY ORDERED k-MEDIAN: We have the cost function $f(\boldsymbol{x}) = \boldsymbol{v}^\top \boldsymbol{x_w}^{\downarrow}$ where the weight vector $\boldsymbol{v} \in \mathbb{R}^n_{\geq 0}$ and the priority vector $\boldsymbol{w} \in \mathbb{R}^P_{\geq 0}$ are given as input and where $\boldsymbol{x_w} = (w(p)x(p))_{p \in P}$. This objective generalizes both PRIORITY k-CENTER and ORDERED k-MEDIAN. Another natural objective is the (multi-level) CASCADED NORM CLUSTERING, which generalizes (z,q)-FAIR CLUSTERING to allow multiple levels of cost aggregation. The cost function f for this problem is described by a directed acyclic graph (DAG) D with one sink node and $|P|$ source nodes (each source corresponds to a point in P). Each non-source node v is associated with a norm ℓ_q for some q, and each edge (u,v) has weight $w_{u,v}$. Given such a DAG D, the value of $f(\boldsymbol{x})$ can be evaluated by computing the evaluations at nodes in $V(D)$ in (topological) order from sources to sink: (i) The evaluation at source $p \in P$ is $\eta(p) = x(p)$, (ii) For any non-source node $v \in V(D)$ labeled with the norm ℓ_q, we evaluate $\eta(v) = \left(\sum_{u \in N^-(v)} w_{u,v} \eta(u)^q\right)^{1/q}$, and (iii) the value of $f(\boldsymbol{x})$ is the evaluation of the sink.

We remark that *asymmetric* norms can potentially make the problem substantially harder. For example, there is a poly-time $\mathcal{O}(1)$-approximation algorithm for symmetric norms [34] but the asymmetric norm makes it $\Omega(\log n / \log\log n)$-hard to approximate even for the special case of ROBUST k-MEDIAN on the line metrics [24].

3.3 Thesis Contribution

In Publication I [3], we answer the following meta question:

> For a given k-clustering objective and a (structured) metric space, does an EPAS exist?

In particular, we identify a combinatorial property of a metric space that we call ϵ-scatter dimension of the metric space and show an algorithmic

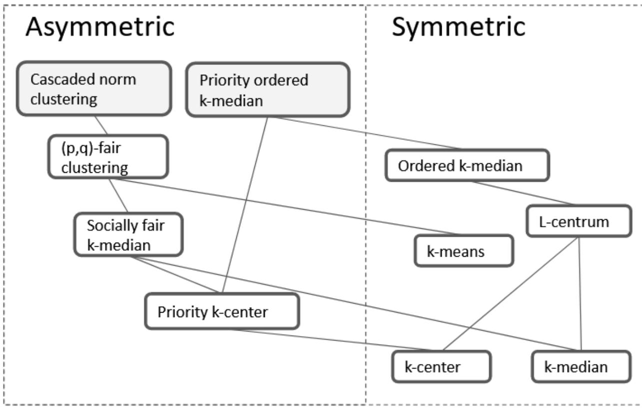

Figure 3.1. Selected clustering objectives that can be formulated as monotone norm minimization. The line illustrates generalization (bottom is a special case of top).

framework that yields an EPAS for several clustering objectives in metrics of bounded ϵ-scatter dimension. Additionally, we show bounds on ϵ-scatter dimension of several metric spaces. We remark that in this chapter, we focus on metric classes that are closed under scaling distances by a constant.

3.3.1 Scattered Dimension

We motivate the notion of ϵ-scatter dimension using the following two player game played in rounds between the *center player* and the *point player* on a metric space $M = (P, F, \delta)$. In each round, the center player chooses a center $x \in F$ such that the unit ball around x covers all the points revealed by the point player. Then, the point player chooses a point $p \in P$ such that $p \notin \mathsf{ball}(x, 1+\epsilon)$. Both players are interested in prolonging the game. The ϵ-scatter dimension of M is the maximum number of rounds in any such game on M.

Definition 3.3.1 (ϵ-Scatter Dimension)**.** *We are given a class $\mathcal{M}$ of finite metric spaces, a space $M = (P, F, \delta)$ in $\mathcal{M}$, and some $\epsilon \in (0,1)$. An* ϵ-scattering *in M is a sequence $(x_1, p_1) \ldots, (x_\ell, p_\ell)$ of center-point pairs $x_i \in F$, $p_i \in P$, $i \in [\ell]$ such that*

$$\delta(x_i, p_j) \le 1 \qquad \text{for all } 1 \le j < i \le \ell \qquad \text{(covering)}$$

$$\delta(x_i, p_i) > 1 + \epsilon \qquad \text{for all } i \in [\ell] \qquad (\epsilon\text{-refutation})$$

The ϵ-scatter dimension *of M is the maximum length of an ϵ-scattering in it. The* ϵ-scatter dimension *of $\mathcal{M}$ is the supremum of the ϵ-scatter dimension over all $M \in \mathcal{M}$.*

We have an immediate observation.

Observation 3.3.1 (ϵ-packing of ϵ-scattering)**.** *Fix* $0 < \epsilon < 1$. *Consider an ϵ-scattering* $(x_1,p_1),\cdots,(x_\ell,p_\ell)$. *Then,* $\{p_1,\cdots,p_\ell\}$ *is an ϵ-packing of* $\mathsf{ball}(x_\ell,1)$. *Similarly,* $\{x_2,\cdots,x_\ell\}$ *is an ϵ-packing of* $\mathsf{ball}(p_1,1)$.

Proof. For the first case, consider p_i,p_j such that $i<j$. Then, $1+\epsilon < d(p_j,x_j) \le d(p_j,p_i)+d(p_i,x_j)$. Thus, $d(p_i,p_j) > \epsilon$ since $d(p_i,x_j) \le 1$. Further, note that $p_i \in \mathsf{ball}(x_\ell,1)$ for every $i \in [\ell]$. Similarly, consider x_i,x_j with $2 \le i < j$. Then, $1+\epsilon < d(p_i,x_i) \le d(p_i,x_j)+d(p_i,P_j)$. Hence, $d(p_i,p_j) > \epsilon$ since $d(p_i,x_j) \le 1$. Additionally, $x_i \in \mathsf{ball}(p_1,1)$ for all $2 \le i \le \ell$. □

Example 3.3.1 (Path on n vertices)**.** *Consider a path P_n on n vertices of length* 2. *Then, observe that ϵ-scatter dimension of P_n is at most* $2/\epsilon$. *This follows from Observation 3.3.1 since $\delta(p_i,p_j) > \epsilon$ in any ϵ-scattering* $(x_1,p_1),\cdots,(x_\ell,p_\ell)$, *for $i \neq j \in [\ell]$. Since the diameter of P_n is* 2, *we have* $\ell \le 2/\epsilon$.

This immediately yields a bound ϵ-scatter dimension of metric of doubling dimension d.

Theorem 3.3.1 (ϵ-scatter dimension of Doubling Metric)**.** *For* $\epsilon \in (0,1)$, *the ϵ-scatter dimension of a metric of doubling dimension d is* $(1/\epsilon)^{\mathcal{O}(d)}$.

Proof. This follows from the fact that ϵ-packing number of a metric of doubling dimension d is at most $(1/\epsilon)^{\mathcal{O}(d)}$ [76]. □

3.3.2 ϵ-scatter Dimension Results

As argued before, in Theorem 3.3.1, we showed that metrics of bounded doubling dimension have bounded ϵ-scatter dimension. Further, in Publication I, we show ϵ-scatter dimension bounds for the following graph metrics.

Theorem 3.3.2. *For* $\epsilon \in (0,1)$, *the ϵ-scatter dimension is* $\exp\left((1/\epsilon)^{\mathcal{O}(\mathsf{tw})}\right)$ *for treewidth-*tw *graphs.*

Theorem 3.3.3. *For* $\epsilon \in (0,1)$, *the ϵ-scatter dimension is* $\exp(\exp(\mathsf{poly}(1/\epsilon)))$ *for planar graphs.*

Further, in Publication I, we relax the notion of ϵ-scatter dimension to what we call *algorithmic ϵ-scatter dimension* for continuous Euclidean space and show that it has bounded algorithmic ϵ-scatter dimension. As before, the notion of bounded algorithmic ϵ-scatter dimension is sufficient to yield an EPAS. We remark that in this thesis, we will not deal with algorithmic ϵ-scatter dimension.

Theorem 3.3.4. *The continuous Euclidean space* (P,F,δ), *that is,* $P \subsetneq \mathbb{R}^d$ *finite, and* $F = \mathbb{R}^d$, *has algorithmic ϵ-scatter dimension* $\mathcal{O}(1/\epsilon^4 \log 1/\epsilon)$.

3.3.3 EPAS Results

Our main algorithmic result is the following.

Theorem 3.3.5. *Let $\mathscr{M}$ be a class of metric spaces that is closed under scaling distances by a positive constant. There is a randomized algorithm that computes for any* NORM k-CLUSTERING *instance $\mathscr{I} = (M, f, k)$ with metric $M = (P, F, \delta) \in \mathscr{M}$, and any $\epsilon \in (0,1)$, with high probability a $(1+\epsilon)$-approximate solution if the following two conditions are met.*

1. *There is an efficient algorithm evaluating for any distance vector $\boldsymbol{x} \in \mathbb{R}^P_{\geq 0}$ the objective $f(\boldsymbol{x})$ in time $T(f)$.*
2. *There exists a function $\lambda \colon \mathbb{R}_+ \to \mathbb{R}_+$, such that for all $\epsilon > 0$, the algorithmic ϵ-scatter dimension of $\mathscr{M}$ is at most $\lambda(\epsilon)$.*

The running time of the algorithm is $\exp\left(\widetilde{\mathscr{O}}\left(\frac{k\lambda(\epsilon/10)}{\epsilon}\right)\right) \cdot \mathsf{poly}(|M|) \cdot T(f)$.

Combining the bounds on ϵ-scatter dimension, we get the following result.

Theorem 3.3.6. *There exists a randomized $2^{h(1/\epsilon)\cdot k \cdot \mathsf{polylog}(k)} \cdot \mathsf{poly}(n)$ time EPAS for* NORM k-CLUSTERING *in the following metrics: (i) metrics of bounded doubling dimension, (ii) continuous Euclidean spaces of any dimension, (iii) bounded treewidth metrics, and (iv) planar metrics.*

3.4 Novelty of Results

Unifying Metric Spaces via Scatter Dimension. The fundamental problems such as k-MEDIAN, k-MEANS and k-CENTER are known to admit efficient parameterized approximation schemes (EPAS) in structured metrics such as Euclidean [14, 90] and graph metrics [86, 57]. However, different clustering objectives require completely different techniques even when restricted to the same metric spaces [14, 90]. Similarly, different EPASes are known for the same objective (such as k-MEDIAN) in different metric spaces that exploit the underlying structure of the respective metric spaces [90, 57].

In our work, we present a clean and simple EPAS framework that is essentially oblivious to the underlying metric space and hence is able to not only unify existing EPASes across different metric spaces but also yield new EPAS for more advanced clustering problems, including PRIORITY k-CENTER, ORDERED k-MEDIAN, and SOCIALLY FAIR k-MEDIAN in metric spaces of bounded ϵ-scatter dimension.

Generalization via NORM k-CLUSTERING. Furthermore, we extend the notion of norm objectives [34] to encompass asymmetric norm objectives, thereby encompassing a wide range of clustering objectives discussed in the literature. This generalization enables us to achieve two significant outcomes: (i) unification of EPASes for multiple clustering objectives (ii) new EPAS for advanced objectives such as SOCIALLY FAIR k-MEDIAN and (z,q)-FAIR CLUSTERING.

3.5 Modelling NORM k-CLUSTERING as SOCIALLY FAIR k-MEDIAN with possibly infinite groups

In this section, we lay the foundational groundwork necessary for the algorithmic framework we are about to introduce for the NORM k-CLUSTERING problem in the following section. Importantly, it is worth noting that our algorithmic framework for NORM k-CLUSTERING is rooted in our novel framework for SOCIALLY FAIR k-MEDIAN, which is a special case of NORM k-CLUSTERING, as mentioned earlier. To establish this connection, in this section, we present an intriguing insight – an instance of the NORM k-CLUSTERING can be modelled as an instance of the SOCIALLY FAIR k-MEDIAN problem with possibly infinite groups. This converse insight enables us to extend the algorithmic framework designed for SOCIALLY FAIR k-MEDIAN to NORM k-CLUSTERING. To begin, we provide the essential background on monotone norms.

Fact 3.5.1. *Any norm is a convex function.*

Definition 3.5.1 (Subgradient)**.** *A subgradient of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ at any point $\boldsymbol{x} \in \mathbb{R}^n$ is any $\boldsymbol{g} \in \mathbb{R}^n$ such that the following holds for every $\boldsymbol{y} \in \mathbb{R}^n$*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}^\top(\boldsymbol{y} - \boldsymbol{x});$$

we denote by $\partial f(\boldsymbol{x})$ the set of subgradients of f at $\boldsymbol{x}$.

The following fact summarizes various useful properties of subgradients specialized to norm functions. Because we apply norm objectives exclusively to non-negative distance vectors, we call (slightly abusing terminology) a restriction of a norm to $\mathbb{R}^n_{\geq 0}$ a norm as well.

Fact 3.5.2 ([34])**.** *Let $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a norm, and $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$. If $\boldsymbol{g}$ is a subgradient of f at $\boldsymbol{x}$, then $f(x) = \boldsymbol{g}^\top \boldsymbol{x}$ and $f(y) \geq \boldsymbol{g}^\top \boldsymbol{y}$ for all $\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$. Further, if f is monotone, there exists a subgradient $\boldsymbol{g} \in \partial f(\boldsymbol{x})$ such that $\boldsymbol{g} \geq 0$.*

The following observation is an immediate consequence of Fact 3.5.2.

Observation 3.5.1. *Let $\partial f = \bigcup_{\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}} \partial f(\boldsymbol{y})$ be the set of all subgradients of f. Then, for any $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$, we have that*

$$f(\boldsymbol{x}) = \max_{\boldsymbol{g} \in \partial f} \boldsymbol{g}^\top \boldsymbol{x}.$$

In this chapter, for the sake of brevity and to ensure the clarity of our ideas, we assume that we can compute exactly a subgradient of the given monotone norm at any point. We remark that Publication I provides a comprehensive analysis for the case where only an approximate oracle for computing subgradients is available.

Connecting norm objective to SOCIALLY FAIR k-MEDIAN. We will make use of the following definition of SOCIALLY FAIR k-MEDIAN that generalizes the earlier definition.

Definition 3.5.2 (SOCIALLY FAIR k-MEDIAN)**.** *Given a clustering space $M = (P, F, \delta)$, a positive integer k, and a collection of m weight vectors $\mathscr{W} = \{\boldsymbol{w} : P \to \mathbb{R}^n_{\geq 0}\}$, the goal is to compute $X \subseteq F, |X| = k$ that minimizes* $\max_{\boldsymbol{w}\in\mathscr{W}} \boldsymbol{w}^\top \boldsymbol{\delta}(P, X)$.

We remark that, in this definition, a group is a weight vector $\boldsymbol{w} \in \mathscr{W} : P \to \mathbb{R}^n_{\geq 0}$. To see that this definition captures the previous definition, note that given a group $S \subseteq P$, we can define the weight vector $\boldsymbol{w}(p) = 1$ if $p \in S$ and 0 otherwise. However, note that this definition generalizes the previous definition since now a point can belong to different groups with different weights.

Lemma 3.5.1 (NORM k-CLUSTERING as SOCIALLY FAIR k-MEDIAN)**.** *Let $I = ((P, F, \delta), k, f)$ be an instance of* NORM k-CLUSTERING. *Then, $J = ((P, F, \delta), k, \mathscr{W} := \partial f)$ is an instance of* SOCIALLY FAIR k-MEDIAN. *Furthermore,*

$$\min_{X\subseteq F, |X|=k} f(\boldsymbol{\delta}(P, X)) = \min_{X\subseteq F, |X|=k} \max_{\boldsymbol{w}\in\mathscr{W}} \boldsymbol{w}^\top \boldsymbol{\delta}(P, X).$$

Proof. First note that ∂f is a collection of non-negative vectors due to Fact 3.5.2. Hence, J is an instance of SOCIALLY FAIR k-MEDIAN. However, note that $\mathscr{W}$ can be possibly an infinite set of weight vectors. Then, using Observation 3.5.1, for a solution $X \subseteq F$ evaluating f at the point $\boldsymbol{\delta}(P, X)$ is precisely the maximum group cost of $\boldsymbol{\delta}(P, X)$ over the groups in $\mathscr{W}$, i.e.,

$$f(\boldsymbol{\delta}(P, X)) = \max_{\boldsymbol{w}\in\mathscr{W}} \boldsymbol{w}^\top \boldsymbol{\delta} P, X.$$

□

This lemma will be useful when we want to extend our algorithm of SOCIALLY FAIR k-MEDIAN to NORM k-CLUSTERING. However, note the non-triviality in designing such extensions since the number of groups in $\mathscr{W}$ is possibly infinite.

3.6 EPAS Framework for Metric Spaces with Bounded Scatter Dimension

In this section, we will design an algorithmic framework for NORM k-CLUSTERING that yields an EPAS for metric spaces that have bounded

ϵ-scatter dimension. We will start with a simple algorithm for k-CENTER, which is the simplest special case of NORM CLUSTERING. Then, we will see a natural extension of the algorithm to PRIORITY k-CENTER (WEIGHTED k-CENTER) and then extend it further nontrivially to WEIGHTED k-MEDIAN. Later, we will extend the algorithm to SOCIALLY FAIR k-MEDIAN (ROBUST k-MEDIAN) and finally modify the algorithm to NORM k-CLUSTERING. We would like to remark that in our algorithms we assume that we know the optimal cost (OPT). This is without loss of generality since we can guess an arbitrarily good approximation of OPT by generating all possible powers of $(1+\epsilon)$ and running our algorithm for each power. This is fine since we are interested in finding a $(1+\epsilon)$-approximate solution. Let $M=(P,F,\delta)$ be the metric space of our interest, and let $\lambda(\epsilon)$ be its ϵ-scatter dimension.

3.6.1 k-CENTER

Consider the following natural algorithm for k-CENTER. We point out that this algorithm is a modification of [14] that later allows us to extend it to other problems of interest.

Algorithm 1: Algorithm for k-CENTER

Data: Instance $\mathscr{I}=(M=(P,F,\delta),k)$ of k-CENTER, error parameter $\epsilon \in (0,1)$, $\mathsf{OPT}>0$, 1-CENTER algorithm $\mathscr{C}$

Result: Solution X of cost at most $(1+\epsilon)\mathsf{OPT}$ if solution of cost at most OPT exists

Let $X=(x_1,\dots,x_k)$ be an arbitrary set of centers;
Let $Q_\kappa=\emptyset$ for $1\le\kappa\le k$;
while $\exists p\in P$ *such that* $\delta(p,X)>(1+\epsilon)\mathsf{OPT}$ **do**
  Pick cluster $\kappa\in[k]$ for p uniformly at random;
  $Q_\kappa \leftarrow Q_\kappa\cup\{p\}$;
  $x_\kappa \leftarrow \mathscr{C}(Q_\kappa,\mathsf{OPT})$ **if** *no x_κ was found* **then fail** ;
end
return X;

In Algorithm 1, we assume that we are given a polynomial-time algorithm $\mathscr{C}$ for 1-CENTER that returns a center that (approximately) covers all input points within distance OPT, if exists. For finite metric spaces, finding a 1-center can be done in linear time, simply by going through all the points in the metric space. For continuous spaces, such as Euclidean, $(1+\epsilon)$-approximate 1-CENTER algorithms are known to exist [13], even for the weighted case [91]. Note that such approximate algorithms are sufficient for our purpose, since we are interested in finding a $(1+\epsilon)$-approximate solution to our problem. However, in this section, for cleaner analysis, we assume that we have an exact algorithm.

Lemma 3.6.1. *Algorithm 1 finds a $(1+\epsilon)$-approximate solution with proba-*

bility $k^{-k\lambda(\epsilon)}$ in time $k\lambda(\epsilon)\mathsf{poly}(n)$.

Proof. First note that if the algorithm terminates successfully, it yields a $(1+\epsilon)$-approximate solution. Fix an optimal solution O. For every $p \in P$ that the algorithm considers in the while loop, the probability that the algorithm correctly picks its optimal cluster is $1/k$. Fix any Q_κ and let $p_1^\kappa, \cdots p_\ell^\kappa$ be the points added by the algorithm to the cluster Q_κ. Let $x_1^\kappa, \cdots, x_\ell^\kappa$ be the cluster centers selected by the algorithm for Q_κ such that x_t^κ is the center returned by $\mathscr{C}(\{p_1^\kappa, \cdots, p_{t-1}^\kappa\}, \mathsf{OPT})$. Assuming the algorithm's random choices for $p_1^\kappa, \cdots p_\ell^\kappa$ are correct, then the sequence $(x_1^\kappa, p_1^\kappa), \cdots, (x_\ell^\kappa, p_\ell^\kappa)$ forms an ϵ-scattering after scaling the distances in M by OPT. Since the ϵ-scatter dimension of M is $\lambda(\epsilon)$, it must be that $\ell \le \lambda(\epsilon)$. Thus, assuming the algorithm's random choices are always correct, the while loop runs for at most $k\lambda(\epsilon)$ times before terminating. Hence, the running time of the algorithm is $k\lambda(\epsilon)\mathsf{poly}(n)$. Finally, the probability that all the random choices of the algorithm are correct is $k^{-k\lambda(\epsilon)}$. □

Hence, repeating Algorithm 1 $\mathcal{O}(k^{k\lambda(\epsilon)})$ times boosts the success probability to a constant close to 1. Thus, in the rest of the chapter, we allow our algorithms to have success probability of the form $1/f(k,\epsilon)$, since this implies an FPT algorithm with constant success probability.

3.6.2 PRIORITY k-CENTER

In PRIORITY k-CENTER, every point p has a weight $w(p)$, and hence a natural extension of Algorithm 1 is as shown in Algorithm 2.

Algorithm 2: Algorithm for PRIORITY k-CENTER (first attempt)

Data: Instance $\mathscr{I} = (M = (P, F, \delta), k)$ of PRIORITY k-CENTER, error parameter $\epsilon \in (0,1)$, $\mathsf{OPT} > 0$, BALL INTERSECTION algorithm $\mathscr{C}$

Result: Solution X of cost at most $(1+\epsilon)\mathsf{OPT}$ if solution of cost at most OPT exists

```
Let X = (x_1, ..., x_k) be an arbitrary set of centers;
Let Q_κ = ∅ for 1 ≤ κ ≤ k;
while ∃p ∈ P such that w(p)δ(p, X) > (1+ε)OPT do
    Pick cluster κ ∈ [k] for p uniformly at random;
    Q_κ ← Q_κ ∪ {(p, OPT/w(p))};
    x_κ ← C(Q_κ) if no x_κ was found then fail ;
end
return X;
```

Following are the changes made.

1. The while condition takes care of weighted distances.
2. Since every $p \in P$ demands a center within a distance $\mathsf{OPT}/w(p)$, Q_κ is now a set of *cluster constraints*.

3. Now we want a BALL INTERSECTION algorithm instead of 1-CENTER since we want to find if there exists a center that satisfies all the balls of cluster constraints in Q_κ. Again, algorithms that find a $(1+\epsilon)$-approximate solution for BALL INTERSECTION are known [91]. But for cleaner analysis, we will work with exact algorithms.

Similar to k-CENTER analysis, let us fix some Q_κ. Let $(p_1^\kappa, r_1^\kappa), \cdots, (p_\ell^\kappa, r_\ell^\kappa)$ be the set of requests made to Q_κ, and let $x_1^\kappa, \cdots, x_\ell^\kappa$ be the corresponding centers (as defined previously). Now suppose there are at most τ different radii in the set of requests. Then, since for each fixed radius R, the corresponding sequence $(x_{R_1}^\kappa, p_{R_1}^\kappa), \cdots, (x_{R_1}^\kappa, p_{R_{\ell'}}^\kappa)$ of points with radius R forms an ϵ-scattering, we have that $\ell \le \tau\lambda(\epsilon)$. For constant τ, we are done since the total number of iterations of the while loop is at most $k\tau\lambda(\epsilon)$, and hence the success probability is $k^{-k\tau\lambda(\epsilon)}$. But if τ is arbitrarily large (say, n^2), then it is not clear that the iterations of the algorithm are bounded. To this end, we introduce the concept of *initial upper bounds* that help us to initialize X cleverly so that τ is always bounded for any Q_κ. First, we need the following definition.

Definition 3.6.1 (ϵ-witness for solution X)**.** *Let X be any solution. We call $p \in P$ an* ϵ-witness *for X is $\delta(p,X) > (1+\epsilon)\delta(p,O)$.*

Initial Upper Bounds. The idea is that suppose in Algorithm 2, the solution X always remains a c-approximate solution, for some $c > 1$. Then, consider any Q_κ, and suppose (p,r) and (p',r') are two requests to Q_κ (in any order) such that $r' \le r$. Then, we claim that $r' \ge \epsilon r/2c$. If not, then consider the iteration when the algorithm added (p,r) to Q_κ, and let X' be the corresponding solution for which p is an ϵ-witness. Now, observe that

$$\delta(p,X') \le \delta(p,p') + \delta(p',X') \le r + r' + cr' < (1 + \epsilon/2c + \epsilon/2)r \tag{3.1}$$

But since $r = \mathsf{OPT}/w(p)$, this implies that $\delta(p,X') < (1+\epsilon)\mathsf{OPT}/w(p)$, which is a contradiction to the fact that p is an ϵ-witness for X'. Hence, our goal is now to make sure that the initial solution X remains c-approximation throughout the course of the algorithm, for some $c > 1$. Algorithm 3 shows the final pseudocode for PRIORITY k-CENTER.

Algorithm 3: Algorithm for PRIORITY k-CENTER

Data: Instance $\mathscr{I} = (M = (P, F, \delta), w : P \to \mathbb{R}_{\geq 0}, k)$ of PRIORITY k-CENTER, error parameter $\epsilon \in (0,1)$, $\mathsf{OPT} > 0$, BALL INTERSECTION algorithm $\mathscr{C}$

Result: Solution X of cost at most $(1+\epsilon)\mathsf{OPT}$ if solution of cost at most OPT exists

Set $u(p) = \mathsf{OPT}/w(p)$ for every $p \in P$;
Sort P in non-decreasing order of $u(p)$;
Mark $p_i \in P$ if $\mathsf{ball}(p_i, u(p_i))$ is disjoint from $\mathsf{ball}(p_j, u(p_j))$ for every $j < i$;
Let $p^{(1)}, \ldots, p^{(k')}$ be the marked points ;
Let $Q_\kappa = \{(p^{(\kappa)}, u(p^{(\kappa)}))\}$ for all $\kappa \in [k']$;
Let $Q_\kappa = \emptyset$ for all κ with $k' < \kappa \leq k$;
Let $X = (x_1, \ldots, x_k)$ be any set of centers where x_κ satisfies the requests in Q_κ;
Let $X = (x_1, \ldots, x_k)$ be an arbitrary set of centers;
Let $Q_\kappa = \emptyset$ for $1 \leq \kappa \leq k$;
while $\exists p \in P$ *such that* $w(p)\delta(p, X) > (1+\epsilon)\mathsf{OPT}$ **do**
    Pick cluster $\kappa \in [k]$ for p uniformly at random;
    $Q_\kappa \leftarrow Q_\kappa \cup \{(p, \mathsf{OPT}/w(p))\}$;
    $x_\kappa \leftarrow \mathscr{C}(Q_\kappa)$ **if** *no* x_κ *was found* **then fail** ;
end
return X;

Observation 3.6.1. *The solution X maintained by Algorithm 3 remains 3-approximation throughout the execution of the algorithm.*

Proof. For point $p \in P$ whose ball is marked, $\delta(p, X) \leq u(p) \leq \mathsf{OPT}/w(p)$ since Q_κ is initialized with the corresponding cluster constraint $(p, u(p))$. Any point $p \in P$ whose ball is not marked means that $\mathsf{ball}(p, u(p))$ intersects with $\mathsf{ball}(p', u(p'))$ such that p' is marked and $u(p') \leq u(p)$. Hence,

$$\delta(p, X) \leq \delta(p, p') + \delta(p', X) \leq u(p) + u(p') + u(p') \leq 3u(p) \leq 3\frac{\mathsf{OPT}}{w(p)}.$$

□

Hence, using Equation 3.1, we have the following crucial lemma.

Claim 3.6.1. *In Algorithm 3, for any Q_κ, the ratio $\frac{r_{max}}{r_{min}} \leq \frac{6}{\epsilon}$.*

Lemma 3.6.2. *Algorithm 3 finds a $(1+\epsilon)$-approximate solution for* PRIORITY k-CENTER *with probability$k^{-\mathcal{O}(\lambda(\epsilon)\frac{k}{\epsilon}\log(\frac{1}{\epsilon}))}$ in time $\mathcal{O}\left(\lambda(\epsilon)\frac{k}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)\mathsf{poly}(n)$.*

Proof. Using Claim 3.6.1, we have that every Q_κ has an aspect ratio of the radius of its requests bounded by $6/\epsilon$. Hence, by standard discretization, there are at most $\mathcal{O}\left(\frac{1}{\epsilon}\log\left(\frac{6}{\epsilon}\right)\right)$ different radii per Q_κ. Since ϵ-scatter dimension of M is bounded by $\lambda(\epsilon)$, we have that Algorithm 3 iterates at most

$\mathcal{O}\left(\lambda(\epsilon)^{\frac{k}{\epsilon}}\log\left(\frac{1}{\epsilon}\right)\right)$ times before terminating. The success probability analysis is as before in Lemma 3.6.1. □

3.6.3 WEIGHTED k-MEDIAN

Let us try to extend Algorithm 3 to WEIGHTED k-MEDIAN. The first obvious change is in the while loop. The condition will be to check if $\sum_{p\in P} w(p)\delta(p,X) > (1+\epsilon)\mathsf{OPT}$ and proceed. Thus, the while loop provides a *linear constraint* while Q_κ requires a cluster constraint, which is about a point whose distance is violated by X. Hence, we need to convert the linear constraint to a cluster constraint using some distance constraint. Fortunately, the following claim provides such a procedure.

Claim 3.6.2. *For any solution X, consider the distribution $\mathcal{D}_X$ on P such that* $\Pr_{p\sim\mathcal{D}_X}[p] := \frac{w(p)\delta(p,X)}{\sum_{p\in P} w(p)\delta(p,X)}$. *Let $W_X = \{p \in P \mid \delta(p,X) > (1+\epsilon/10)\delta(p,O)\}$. Then, if $\sum_{p\in P} w(p)\delta(p,X) > (1+\epsilon)\mathsf{OPT}$ then $\Pr_{p\sim\mathcal{D}_X}[p \in W_X] \geq \frac{\epsilon}{100}$.*

Proof. Suppose $\sum_{p\in P} w(p)\delta(p,X) > (1+\epsilon)\mathsf{OPT}$, but assume for contradiction that $\Pr_{p\sim\mathcal{D}_X}[p \in W_X] < \frac{\epsilon}{100}$, and hence $\Pr_{p\sim\mathcal{D}_X}[p \notin W_X] \geq 1-\frac{\epsilon}{100}$. Then, we have

$$\Pr_{p\sim\mathcal{D}_X}[p \notin W_X] = \sum_{a\notin W_X} \frac{w(a)\delta(a,X)}{\sum_{p\in P} w(p)\delta(p,X)} \geq 1-\frac{\epsilon}{100}.$$

Hence,

$$\left(1+\frac{\epsilon}{10}\right)\sum_{a\notin W_X} w(a)\delta(a,O) \geq \sum_{a\notin W_X} w(a)\delta(a,X) \geq \left(1-\frac{\epsilon}{100}\right)\sum_{p\in P} w(p)\delta(p,X)$$

Thus, we have

$$\sum_{p\in P} w(p)\delta(p,X) \leq \frac{1+\epsilon/10}{1-\epsilon/100}\sum_{a\notin W_X} w(a)\delta(a,O) \leq \left(1+\frac{2\epsilon}{5}\right)\mathsf{OPT},$$

which is a contradiction since we assumed $\sum_{p\in P} w(p)\delta(p,X) > (1+\epsilon)\mathsf{OPT}$. □

The next problem with Algorithm 3 is that we want to compute upper bounds for WEIGHTED k-MEDIAN that guarantee a bounded aspect ratio for radii for any Q_κ.

Initial Upper Bounds. For $p \in P$ and $r \in \mathbb{R}_+$, let $w(\mathsf{ball}(p,r))$ denote the total weight of points in $\mathsf{ball}(p,r)$.

Observation 3.6.2. *For $p \in P$ and $r \in \mathbb{R}_+$, if $w(\mathsf{ball}(p,r)) > \frac{\mathsf{OPT}}{r}$ then $\mathsf{ball}(p,2r) \cap O \neq \emptyset$.*

Proof. Suppose $\mathsf{ball}(p,2r) \cap O = \emptyset$, then we have $\sum_{p'\in\mathsf{ball}(p,r)} w(p')\delta(p',O) > r \cdot \frac{\mathsf{OPT}}{r} = \mathsf{OPT}$. □

Hence, $u(p) := 2 \cdot \min\{r \mid w(\mathsf{ball}(p,r)) > \mathsf{OPT}/r\}$ is a feasible upper bound on $\delta(p,O)$. In fact, with $u(p)$, we can prove a stronger version of Claim 3.6.2.

Claim 3.6.3. *For $p \in P$, let $u(p) = 2 \cdot \min\{r \mid w(\mathsf{ball}(p,r)) > \mathsf{OPT}/r\}$. For a solution X, let $\mathcal{D}_X$ and W_X be as defined in Claim 3.6.2. Let $A_X = \{p \in P \mid \delta(p,X) \geq \frac{\epsilon}{4k} u(p)\}$. Then, if $\sum_{p \in P} w(p)\delta(p,X) > (1+\epsilon)\mathsf{OPT}$, then $\Pr_{p \sim \mathcal{D}_X}[p \in W_X \mid p \in A_X] \geq \frac{\epsilon}{1000}$.*

The proof of the above claim is technical and can be found in Publication I. Algorithm 4 shows the final pseudocode for WEIGHTED k-MEDIAN.

Algorithm 4: Algorithm for WEIGHTED k-MEDIAN

Data: Instance $\mathcal{I} = (M = (P,F,\delta), w : P \to \mathbb{R}_{\geq 0}, k)$ of WEIGHTED k-MEDIAN, error parameter $\epsilon \in (0,1)$, $\mathsf{OPT} > 0$, BALL INTERSECTION algorithm $\mathcal{C}$

Result: Solution X of cost at most $(1+\epsilon)\mathsf{OPT}$ if solution of cost at most OPT exists

For each $p \in P$, compute $u(p) = 2 \cdot \min\{r \mid w(\mathsf{ball}(p,r)) > \mathsf{OPT}/r\}$;
Sort P in non-decreasing order of $u(p)$;
Mark $p_i \in P$ if $\mathsf{ball}(p_i, u(p_i))$ is disjoint from $\mathsf{ball}(p_j, u(p_j))$ for every $j < i$;
Let $p^{(1)}, \ldots, p^{(k')}$ be the marked points ;
Let $Q_\kappa = \{(p^{(\kappa)}, u(p^{(\kappa)}))\}$ for all $\kappa \in [k']$;
Let $Q_\kappa = \emptyset$ for all κ with $k' < \kappa \leq k$;
Let $X = (x_1, \ldots, x_k)$ be any set of centers where x_κ satisfies the requests in Q_κ;
Let $X = (x_1, \ldots, x_k)$ be an arbitrary set of centers;
Let $Q_\kappa = \emptyset$ for $1 \leq \kappa \leq k$;
while $\sum_{p \in P} w(p)\delta(p,X) > (1+\epsilon)\mathsf{OPT}$ **do**
  $A_X \leftarrow \left\{ p \in P \mid \delta(p,X) \geq \frac{\epsilon u(p)}{1000k} \right\}$;
  Sample an element $p \in A_X$ where $\mathbf{Pr}[p = a] = \frac{w(a)\delta(a,X)}{\sum_{b \in A_X} w(b)\delta(b,X)}$ for any $a \in A_X$;
  Pick cluster $\kappa \in [k]$ for p uniformly at random;
  $Q_\kappa \leftarrow Q_\kappa \cup \{(p, \frac{\delta(p,X)}{(1+\epsilon/10)})\}$;
  $x_\kappa \leftarrow \mathcal{C}(Q_\kappa)$ **if** *no x_κ was found* **then fail** ;
end
return X;

Lemma 3.6.3. *Algorithm 4 finds a $(1+\epsilon)$-approximate solution for* WEIGHTED k-MEDIAN *with probability $\exp\left(-\tilde{\Omega}\left(\lambda\left(\frac{\epsilon}{10}\right)\frac{k}{\epsilon}\right)\right)$ in time $\mathcal{O}\left(\lambda\left(\frac{\epsilon}{10}\right)\frac{k}{\epsilon}\log\left(\frac{k}{\epsilon}\right)\right)\mathsf{poly}(n)$.*

Proof. As before, it is sufficient to bound the aspect ratio of the radii of requests to any Q_κ. Consider any Q_κ, and suppose (p,r) and (p',r') are two requests to Q_κ (in any order) such that $r' \leq r$. Then, we claim that $r' \geq \epsilon^2 r/10^5 k$. If not, then consider the iteration when the algorithm added

(p,r) to Q_κ, and let X' be the corresponding solution for which p is an $\epsilon/10$-witness. First, note that $r \geq \epsilon u(p)/1000k$ and $r' \geq \epsilon u(p')/1000k$. We have,

$$\delta(p,X') \leq \delta(p,p') + \delta(p',X') \leq r + r' + 3u(p') \leq r + \left(1 + \frac{3000k}{\epsilon}\right) r' < \left(1 + \frac{\epsilon}{10}\right) r.$$

But this is a contradiction since $r = \frac{d(p,X')}{1+\epsilon/10}$.

Hence, we have that every Q_κ has an aspect ratio of the radius of its requests bounded by $10^5 k/\epsilon^2$. Thus, using standard discretization and ϵ-scatter dimension of M, we have that Algorithm 4 iterates at most $\mathcal{O}\left(\lambda\left(\frac{\epsilon}{10}\right)\frac{k}{\epsilon}\log\left(\frac{k}{\epsilon}\right)\right)$ times before terminating. Using Claim 3.6.3, the success probability is $(k/\epsilon)^{-\Omega\left(\lambda\left(\frac{\epsilon}{10}\right)\frac{k}{\epsilon}\log\left(\frac{k}{\epsilon}\right)\right)}$. $\square$

3.6.4 SOCIALLY FAIR k-MEDIAN

Now consider the problem SOCIALLY FAIR k-MEDIAN, and let $\mathscr{I} = (M = (P,F,\delta), w : P \to \mathbb{R}_{\geq 0}, k, \mathscr{S} = \{S_1, \cdots, S_m\})$ be an instance of SOCIALLY FAIR k-MEDIAN. Recall that the group cost of a group S_i for a solution $X \subseteq F, |X| = k$ is defined as $\sum_{p \in S_i} w(p)\delta(p,X)$, and the cost of solution X is the maximum group cost for X. The objective of SOCIALLY FAIR k-MEDIAN is to find a solution $X \subseteq F$ of size k that has the minimum cost. Formally, we want to compute $\arg\min_{X \subseteq F, |X|=k} \max_{i \in [m]} \sum_{p \in S_i} w(p)\delta(p,X)$. Hence, the definition of this problem resembles both k-CENTER and k-MEDIAN– we are interested in minimizing the maximum group cost similar to k-CENTER, while the group cost is defined as the k-MEDIAN cost of the points in the group. Thus, we can modify Algorithm 4 accordingly to obtain Algorithm 5 that has the following changes.

1. **Upper Bounds.** Observe that SOCIALLY FAIR k-MEDIAN behaves as a collection of WEIGHTED k-MEDIAN instances that correspond to the groups. Hence, to compute the upper bound $u(p)$ for $p \in P$, we need to consider the group costs. More precisely,

$$u(p) := 2 \times \min_{r \in \mathbb{R}} \max_{S \in \mathscr{S}} w_S(\mathsf{ball}(p,r)) > \frac{\mathsf{OPT}}{r},$$

   where $w_S(\mathsf{ball}(p,r)) := \sum_{p' \in \mathsf{ball}(p,r) \cap S} w(p)$.

2. **While condition.** In this case, we are interested in the group that violates the present solution X.

3. **Sampling.** Similarly, we sample an ϵ-witness from the violated group according to the weight of points in that group.

The proof of the following lemma follows a similar line of reasoning to that of Lemma 3.6.3.

Lemma 3.6.4. *With probability* $\exp\left(-\tilde{\Omega}\left(\lambda\left(\frac{\epsilon}{10}\right)\frac{k}{\epsilon}\right)\right)$*, Algorithm 5 finds a* $(1+\epsilon)$*-approximate solution for* SOCIALLY FAIR k-MEDIAN *in time* $\mathcal{O}\left(\lambda\left(\frac{\epsilon}{10}\right)\frac{k}{\epsilon}\log\left(\frac{k}{\epsilon}\right)\right)\mathsf{poly}(n)$.

Algorithm 5: Algorithm for SOCIALLY FAIR k-MEDIAN

Data: Instance $\mathcal{I} = (M = (P, F, \delta), w : P \to \mathbb{R}_{\geq 0}, k, \mathcal{S} = \{S_1, \cdots, S_m\})$, error parameter $\epsilon \in (0,1)$, $\mathsf{OPT} > 0$, BALL INTERSECTION algorithm $\mathcal{C}$

Result: Solution X of cost at most $(1+\epsilon)\mathsf{OPT}$ if solution of cost at most OPT exists

For each $p \in P$, compute $u(p) = 2 \cdot \min\{r \mid \max_{S \in \mathcal{S}} w_S(\mathsf{ball}(p, r)) > \mathsf{OPT}/r\}$;
Sort P in non-decreasing order of $u(p)$;
Mark $p_i \in P$ if $\mathsf{ball}(p_i, u(p_i))$ is disjoint from $\mathsf{ball}(p_j, u(p_j)$ for every $j < i$;
Let $p^{(1)}, \ldots, p^{(k')}$ be the marked points ;
Let $Q_\kappa = \{(p^{(\kappa)}, u(p^{(\kappa)}))\}$ for all $\kappa \in [k']$;
Let $Q_\kappa = \emptyset$ for all κ with $k' < \kappa \leq k$;
Let $X = (x_1, \ldots, x_k)$ be any set of centers where x_κ satisfies the requests in Q_κ;
Let $X = (x_1, \ldots, x_k)$ be an arbitrary set of centers;
Let $Q_\kappa = \emptyset$ for $1 \leq \kappa \leq k$;
while $\exists S \in \mathcal{S}$ *such that* $\sum_{p \in S} w(p)\delta(p, X) > (1+\epsilon)\mathsf{OPT}$ **do**
  $A_S \leftarrow \left\{ p \in S \mid \delta(p, X) \geq \frac{\epsilon u(p)}{1000k} \right\}$;
  Sample an element $p \in A_S$ where $\mathbf{Pr}[p = a] = \frac{w(a)\delta(a,X)}{\sum_{b \in A_S} w(b)\delta(b,X)}$ for any $a \in A_S$;
  Pick cluster $\kappa \in [k]$ for p uniformly at random;
  $Q_\kappa \leftarrow Q_\kappa \cup \{(p, \frac{\delta(p,X)}{(1+\epsilon/10)})\}$;
  $x_\kappa \leftarrow \mathcal{C}(Q_\kappa)$ **if** *no* x_κ *was found* **then fail** ;
end
return X;

3.6.5 NORM k-CLUSTERING

In the final section, we will extend Algorithm 5 to NORM k-CLUSTERING. Recall that in NORM k-CLUSTERING, we are given an instance $\mathcal{I} = (M = (P, F, \delta), k, f : \mathbb{R}^P_{\geq 0} \to \mathbb{R}_{\geq 0})$, where f is a monotone norm, and we want to compute $\arg\min_{X \subseteq F, |X| = k} f(\boldsymbol{\delta}(P, X))$, where $\boldsymbol{\delta}(P, X) := (\delta(p, X))_{p \in P}$. The idea is to use the result of Lemma 3.5.1 that allows us to treat an instance of NORM k-CLUSTERING as an instance of SOCIALLY FAIR k-MEDIAN with a possibly infinite number of groups. Thus, if we can extend our Algorithm 5 to the case where the number of groups can be possibly infinite, then we can also compute a $(1+\epsilon)$-approximate solution for NORM k-CLUSTERING. This is precisely the idea of Algorithm 6.

Algorithm 6: Algorithm for NORM k-CLUSTERING

Data: Instance $\mathscr{I} = (M = (P, F, \delta), k, f : \mathbb{R}^P_{\geq 0} \to \mathbb{R}_{\geq 0}$ of NORM k-CLUSTERING, error parameter $\epsilon \in (0,1)$, $\mathsf{OPT} > 0$, BALL INTERSECTION algorithm $\mathscr{C}$

Result: Solution X of cost at most $(1+\epsilon)\mathsf{OPT}$ if solution of cost at most OPT exists

For each $p \in P$, compute $u(p) = 2 \cdot \min\{\alpha > 0 \mid f(\mathbf{1}_{p,\alpha}) > \mathsf{OPT}/\alpha\}$;
Sort P in non-decreasing order of $u(p)$;
Mark $p_i \in P$ if $\mathsf{ball}(p_i, u(p_i))$ is disjoint from $\mathsf{ball}(p_j, u(p_j)$ for every $j < i$;
Let $p^{(1)}, \ldots, p^{(k')}$ be the marked points ;
Let $Q_\kappa = \{(p^{(\kappa)}, u(p^{(\kappa)}))\}$ for all $\kappa \in [k']$;
Let $Q_\kappa = \emptyset$ for all κ with $k' < \kappa \leq k$;
Let $X = (x_1, \ldots, x_k)$ be any set of centers where x_κ satisfies the requests in Q_κ;
Let $X = (x_1, \ldots, x_k)$ be an arbitrary set of centers;
Let $Q_\kappa = \emptyset$ for $1 \leq \kappa \leq k$;
while $f(\boldsymbol{\delta}(P, X)) > (1+\epsilon)\mathsf{OPT}$ **do**
    $\boldsymbol{w} \leftarrow$ subgradient of f at $\boldsymbol{\delta}(P, X)$;
    $A_X \leftarrow \left\{ p \in P \mid \delta(p, X) \geq \frac{\epsilon u(p)}{1000k} \right\}$;
    Sample an element $p \in A_X$ where $\mathbf{Pr}[p = a] = \frac{w(a)\delta(a,X)}{\sum_{b \in A_X} w(b)\delta(b,X)}$ for any $a \in A_X$;
    Pick cluster $\kappa \in [k]$ for p uniformly at random;
    $Q_\kappa \leftarrow Q_\kappa \cup \{(p, \frac{\delta(p,X)}{(1+\epsilon/10)})\}$;
    $x_\kappa \leftarrow \mathscr{C}(Q_\kappa)$ **if** *no* x_κ *was found* **then fail** ;
end
return X;

It requires handling two issues of Algorithm 5.

1. **Upper Bounds.** Observe that in SOCIALLY FAIR k-MEDIAN, we were interested in computing the maximum weight of a group in a given ball. Even though we now have a possibly infinite number of groups, we can still compute a group (subgradient) that has the maximum weight in a given ball when given access to the evaluation oracle for f. Let $\mathbf{1}_{p,\alpha}$ be the characteristic vector of points in $\mathsf{ball}(p, \alpha)$. Then, using Observation 3.5.1, note that

$$\max_{\boldsymbol{g} \in \partial f} w_{\boldsymbol{g}}(\mathsf{ball}(p, \alpha)) = \max_{\boldsymbol{g} \in \partial f} \boldsymbol{g}^\top \mathbf{1}_{p,\alpha} = f(\mathbf{1}_{p,\alpha}).$$

2. **While condition (Violating group).** In this case, we are interested in a group that violates the present solution X. Using the same arguments as above, we have, for a given distance vector $\boldsymbol{\delta}(P, X)$,

$$\max_{\boldsymbol{g} \in \partial f} \boldsymbol{g}^\top \boldsymbol{\delta}(P, X) = f(\boldsymbol{\delta}(P, X)).$$

We have the main theorem of this section as follows.

Theorem 3.6.1. *Algorithm 6 finds a* $(1+\epsilon)$*-approximate solution for* SOCIALLY FAIR k-MEDIAN *with probability* $\exp\left(-\tilde{\Omega}\left(\lambda\left(\frac{\epsilon}{10}\right)\frac{k}{\epsilon}\right)\right)$ *in time* $\mathcal{O}\left(\lambda\left(\frac{\epsilon}{10}\right)\frac{k}{\epsilon}\log\left(\frac{k}{\epsilon}\right)\right)\mathit{poly}(n)$.

3.7 Conclusion and Open Problems

In this chapter, we have introduced and defined a combinatorial property of metric spaces known as the ϵ-scatter dimension. Subsequently, we have shown that a small ϵ-scatter dimension is sufficient for obtaining an EPAS for NORM k-CLUSTERING by designing an algorithmic framework that is essentially oblivious to the underlying metric space and yet is able to leverage the inherent structure of the metric space. The strength of our framework lies in its simplicity and elegance, as it does not rely on coreset construction. Moreover, it provides a unified approach that encompasses existing EPASes for various clustering objectives in different metric spaces. Furthermore, our framework yields new EPASes for advanced objectives in diverse metric spaces.

Our result opens up several intriguing avenues for further research. One interesting direction is to explore the ϵ-scatter dimension of other graph metrics, such as minor-free graphs. Additionally, improving the ϵ-scatter dimension of bounded treewidth graphs is of independent interest. In terms of the algorithmic framework, it would be fascinating to extend it to clustering objectives that go beyond norms. For instance, many objectives impose specific constraints on how points are assigned to open centers, such as capacity [6, 61, 50], different notions of fairness [21, 46, 45], and diversity constraints [97, 122, 121]. These extensions present exciting opportunities for future investigation.

4. Clustering with Fairness and Robustness Constraints

In the last chapter, we saw that the Norm k-Clustering problem is quite general, and it captures clustering objectives that range from simple and classical ones, such as k-Median, k-Means, k-Center to more complex clustering objectives, such as Socially Fair k-Median (Robust k-Median) and (z,q)-Fair Clustering. Further, we designed an EPAS framework for all these problems under metric spaces that have bounded ϵ-scatter dimension. More specifically, in Publication I, we designed an EPAS for Norm k-Clustering for the *continuous* Euclidean metric space ($F = \mathbb{R}^d$), using a relaxed notion of ϵ-scatter dimension, called *algorithmic ϵ-scatter dimension*, and showing that high dimensional continuous Euclidean metric space has bounded algorithmic ϵ-scatter dimension. However, it turns out that the algorithmic ϵ-scatter dimension (and hence ϵ-scatter dimension) of the finite discrete Euclidean space ($F \subset \mathbb{R}^d$) is unbounded in ϵ, and hence the framework of Publication I yields an EPAS for this setting only when $d = o(\log\log n)$. In this chapter, we ask the question: is this result tight, or does there exist an EPAS for larger dimensions? In fact, our goal in this chapter, that is based on Publication II [2], is quite broader – studying the complete parameterized approximability of a more general problem in a more general metric space.

> What exactly is the parameterized approximability of Robust (k,z)-Clustering in doubling metrics with respect to parameter k?

Further, we also consider Robust (k,z)-Clustering in high dimensional finite discrete Euclidean space and show better approximation factors in FPT time, bypassing the lower bounds of general metrics.

4.1 Algorithmic Fairness and Robustness

Algorithmic fairness, particularly fair clustering [1], has received significant attention from the research community in the past half-decade. The high-level goal here is to guarantee some notion of fairness in the chosen solution. In clustering, there are at least 10 notions of fairness (see [1] for a survey and references therein). For example, [122] considers fairness in the facilities opened by avoiding under-representation and over-representation of communities (this is the topic of our next chapter). Similarly, there are notions of fairness in the clusters given by an algorithm, such as Socially Fair, Individually Fair, and Proportionally Fair, to name a few. In this chapter, we consider the SOCIALLY FAIR (k,z)-CLUSTERING problem, recently introduced independently by Abbasi et al. [4] and Ghadiri et al. [70]. Here, we are given a clustering space (P,F,δ) and m groups $S_1,\dots,S_m$ where each group $S_i \subseteq P$, and our goal is to open k facilities in F, $X \subseteq F, |X| = k$, that minimizes the maximum cost of any group, where the cost of group S_i for X is the sum of weighted distances to power z of points in S_i to X. Thus, in some sense, the objective here is to find a solution that is socially fair for the groups.

Another notion that has also been studied extensively in the algorithmic community is the notion of robustness. The objective here, for example, is to find a solution that is robust under uncertainty. For instance, in clustering, the ROBUST (k,z)-CLUSTERING problem [8] is defined when there is uncertainty in the input data set, and the goal is to find a solution that is robust to these uncertainties. Mathematically, in this problem, we are given a clustering space (P,F,δ) and m scenarios $S_1,\dots,S_m$ where $S_i \subseteq F$, that represent uncertainty in the input data, and the goal is to open k facilities in F, $X \subseteq F, |X| = k$, that minimizes the maximum cost of any scenario, where the cost of scenario S_i for X is the sum of weighted distances to power z of points in S_i to X. Note the intriguing similarity between the definition of this problem and that of SOCIALLY FAIR (k,z)-CLUSTERING– these are mathematically the same problems – however, they originate in completely different contexts.

4.1.1 Problem Definition

Definition 4.1.1 (ROBUST (k,z)-CLUSTERING). *Given a metric space (P,F,δ) with $n = |P|$, positive integer k, a weight function $w : P \to \mathbb{R}_{\geq 0}$, and m groups $S_1,\dots,S_m$ such that $S_i \subseteq P, P = \cup_{i\in[m]} S_i$, find a k-element subset $X \subseteq F$ that minimizes* $\max_{i\in[m]} \sum_{p\in S_i} w(p)\delta(p,X)^z$.

We remark that, in addition to generalizing k-MEDIAN and k-MEANS, the ROBUST (k,z)-CLUSTERING problem encapsulates k-CENTER, when each group contains a distinct singleton.

Let $\mathscr{I} = ((P,F,\delta),k,\{S_1,\dots,S_m\})$ be an instance of ROBUST (k,z)-CLUSTERING. Let $X \subseteq F$. We denote by $\mathsf{cost}(S_i,X)$ to be the (k,z)-CLUSTERING cost of group S_i, i.e., $\mathsf{cost}(S_i,X) := \sum_{p\in S_i} \boldsymbol{w}_p \delta(p,X)^z$. Similarly, we denote by $\mathsf{cost}(\mathscr{I},X)$ to be the ROBUST (k,z)-CLUSTERING cost of $\mathscr{I}$, i.e., $\mathsf{cost}(\mathscr{I},X) := \max_{i\in[m]} \mathsf{cost}(S_i,X)$.

4.1.2 Continuous and Finite Discrete Euclidean Space

For clustering problems in Euclidean space, it *does* matter what restrictions we put on the set of facilities F. For example, when $F = \mathbb{R}^d$, called the *continuous* case, k-MEANS is **NP**-hard even on a plane [125]; further, k-MEANS is **NP**-hard in Euclidean space even when $k = 2$ [7]. Similarly, for polynomial time approximations, continuous k-MEDIAN is **NP**-hard to approximate better than 2 [54], while for the *discrete* case when $F \subset \mathbb{R}^d$, the best hardness of approximation is the factor $1 + 1/e$. Thus, the polynomial time literature suggests that the continuous case is harder than the discrete case. Surprisingly, in this chapter, we will show that, for FPT time the opposite is true. Specifically, in the last chapter, we designed an EPAS for continuous ROBUST (k,z)-CLUSTERING, while in this chapter, we will show that the discrete counter part is hard to approximate, and hence it does not admit an EPAS unless $\mathbf{W[1]} = \mathsf{FPT}$. We would like to remark that, sometimes, we refer to finite discrete Euclidean space simply as discrete Euclidean space.

4.1.3 Related Work

For polynomial time, Makarychev and Vakilian [103] design a $\mathcal{O}(\log m/\log\log m)$-approximation algorithm, which is tight under a standard complexity assumption [24] (In fact, the hardness result holds for ROBUST k-MEDIAN even on a line metric). Hence, the recent focus of the algorithmic community has been obtaining parameterized (FPT) approximation algorithms with better factors. In this direction, an FPT time $(3^z + \epsilon)$-approximation algorithm has been proposed and shown to be tight under Gap-ETH [72]. When parameterized by the number of groups m (instead of k), Ghadiri et al. designed a $(5+2\sqrt{6}+\epsilon)^2$-approximation algorithm in $n^{\mathcal{O}(m^2)}$ time [71].

4.2 Thesis Contribution

In Publication II, we answer the question asked in the introduction of this chapter by making significant progress on the parameterized approximability of ROBUST (k,z)-CLUSTERING in discrete spaces. First, we show an EPAS for the doubling metric of the sub-logarithmic dimension as follows.

Theorem 4.2.1 (EPAS for Doubling Metric of sub-logarithmic dimension)**.** *There is an algorithm that, given an instance of* ROBUST (k,z)-CLUSTERING *in a metric of doubling dimension d, computes a $(1+\epsilon)$-approximate solution for the instance, for every $\epsilon > 0$, in time $f(k,d,\epsilon,z)\mathsf{poly}(m,n)$, where $f(k,d,\epsilon,z) = \left(\left(\frac{2^z}{\epsilon}\right)^d k\log k\right)^{\mathcal{O}(k)}$.*

Next, we complement the above result with a tight hardness result for $\mathcal{O}(\log n)$ dimension Euclidean space.

Theorem 4.2.2 (Hardness of Discrete Euclidean). *For any constant positive integer q, it is* **W[1]**-*hard to approximate the discrete k-CENTER problem in $\mathbb{R}^{O(\log n)}$ under the ℓ_q metrics to within a factor of* $(3/2-\eta)^{1/q}$ *for any positive constant* $\eta > 0$.

Finally, we consider the high dimensional discrete case and show that it is possible to beat the lower bound of $3^z - o(1)$ for ROBUST (k,z)-CLUSTERING in general metrics.

Theorem 4.2.3 (High Dimensional Setting). *There exists a universal constant* $\eta_0 > 0.0006$ *such that for any positive integer z, there is a factor* $3^z(1-\eta_0)$ *FPT approximation algorithm for* ROBUST (k,z)-CLUSTERING *in discrete $\mathbb{R}^d$ that runs in time* $2^{\mathcal{O}_z(k\log k)}\mathsf{poly}(m,n,d)$.

In this chapter, we will prove a slightly weaker version of Theorem 4.2.1, which captures all the essential ideas of the algorithm of Theorem 4.2.1. We remark that these ideas are not only clean to present but are also sufficient to get the result of Theorem 4.2.1.

4.3 Novelty of Results

4.3.1 EPAS for Finite Discrete Geometric Spaces

Our algorithm consists of two main components, both based on the standard techniques from the literature: instance compression and decomposition of the doubling metric into smaller balls. However, it turns out that a natural construction for these standard techniques for ROBUST (k,z)-CLUSTERING faces serious information theoretic limitations, as argued next. A natural idea to compress a ROBUST (k,z)-CLUSTERING instance is to reduce the number of groups since each group can be further compressed using (k,z)-CLUSTERING coreset (such coresets exist [58]), and hence this yields a much smaller instance. Suppose we can reduce the number of groups to $m' \ll m$, while approximately preserving the cost for every solution, then we can obtain an EPAS as follows. First, use a (k,z)-CLUSTERING coreset on every group of the compressed instance to obtain another instance of ROBUST (k,z)-CLUSTERING that has m' groups, each with $g(k,\epsilon)$ points, for some function g. Note that this compression is fine for obtaining an EPAS since the coreset of a group approximately preserves the (k,z)-CLUSTERING cost of the group. Next, enumerate all k-partitions of the points of every group to find a potential solution, and finally, return the solution that has the minimum ROBUST

(k,z)-CLUSTERING cost. Unfortunately, since ROBUST (k,z)-CLUSTERING captures k-CENTER and hence faces a coreset lower bound of $2^{\Omega(d)}$ in Euclidean space of dimension d [16] on the number of new groups (and thus, $m' \geq 2^{\Omega(d)}$). Hence, the running time of this algorithm is $k^{2^{\Omega(d)}}\mathsf{poly}(n,m)$, which is doubly exponential in d. But now, note that this algorithm matches the running time of our algorithm from the last chapter (Publication I), and hence does not yield an EPAS for sub-logarithmic dimension.

Further, note that it is not even clear how to extend the coreset of k-CENTER to reduce the number of groups in an instance of ROBUST (k,z)-CLUSTERING since, first, we would need a mapping between the old groups and new groups, and second, it would be the bare minimum requirement that such mapping should approximately preserve the ROBUST (k,z)-CLUSTERING cost for every solution. Another possibility for compressing the instance would be to reduce the number of points in P instead of the groups and hope to design an EPAS that can exploit small P (without bothering about the number of groups). Again, for this approach to work, it is necessary to have a bijection between the old and the new groups. But it is not clear if such a bijection exists because, in typical coreset constructions, every point in the coreset P' (of P) has a weight that is the sum of the weights of the points in its local neighborhood in P, that it is supposed to represent in P'. However, these points in P could possibly belong to different groups, making it harder to map the groups.

The key idea of our approach is to work with an alternate and general definition of groups that allows a point to participate in different groups with different weights. In this new definition, instead of considering groups as subsets of points, we consider each group as a weight function from points to non-negative reals. This allows different weights to be assigned to the same point by different groups (which can be of practical interest!). Using this new definition, we can think of an approach that compresses the points such that each point in the compressed instance can have a weight for group g that is sum of the weights of nearby points in g that got filtered out in compression. Thus, in a sense, this allows us to approximately preserve the group costs. With this approach and some more technical work that exploits the standard ball decomposition technique of doubling metrics, we obtain a coreset for ROBUST (k,z)-CLUSTERING that can be used to design an EPAS for doubling metrics of sub-logarithmic dimension.

4.3.2 Hardness

Our proof constructs an instance of the discrete k-CENTER from an instance of MULTI-COLORED INDEPENDENT SET problem, which is known to be $W[1]$-hard. In MULTI-COLORED INDEPENDENT SET, we are given a k-partite graph G with a k-partition of the vertices $V_1,\ldots,V_k$, and the goal is to determine if there is an independent set that contains precisely one node from each set V_i, $i \in [k]$. The gadget in our construction is a set of nearly equidistant binary code words. Such code words with relative Hamming distance of roughly 1/2 and logarithmic length are known to exist (see Ta-Shma [119]). The high level idea is as follows. We associate each vertex of G a unique code word of suitable length t. Then, we generate a data point in P for each vertex and edge of G by using the code word(s) associated with the corresponding vertices. The construction guarantees the following crucial properties: (i) The Hamming distance between the data points of vertices is roughly t. (ii) The Hamming distance between a data point of vertex $v \in V_i$ and a data point of an edge e is roughly t if e is incident on $V_i \setminus \{v\}$ and is roughly $3t/2$ otherwise. (iii) The Hamming distance between the data points of the edges is at least (close to) $3t/2$. Thus, the construction forces us to we pick data points of vertices as centers in our solution and guarantees that the optimum cost of the k-CENTER instance is roughly t if and only if there is an independent set in G. As a result, approximating the cost of the k-CENTER instance better than a (roughly) $(3/2)^{1/q}$ factor would imply $W[1] = FPT$. That is because the cost of a k-CENTER instance is the maximum ℓ_q distance between a data point and its closest selected center, and hence, approximating this cost better than the mentioned factor allows us to distinguish between YES and NO cases of an arbitrary instance of MULTI-COLORED INDEPENDENT SET.

4.3.3 High dimensional Discrete Euclidean Space

We build on an idea from [72] that gives a factor-$(3^z + \epsilon)$ FPT approximation for the ROBUST (k,z)-CLUSTERING problem in general metric. Their result crucially utilizes a simple property of metric spaces, which we call the “projection lemma”: Let (Y,δ) be a metric space, and $B \subseteq Y$. Then for any set $X \subseteq Y$, there exists an assignment $\sigma: X \to B$ such that, for all $y \in Y$, we have $\delta(y,\sigma(X)) \leq 2\delta(y,X) + \delta(y,B)$.

Intuitively, this lemma allows us to “project” any set X to $\sigma(X) \subseteq B$ in a way that the distance $\delta(y,\sigma(X))$ can be “charged to” $\delta(y,X)$ and $\delta(y,B)$. Goyal and Jaiswal [72] show how to use this lemma to devise a $(3^z + \epsilon)$ FPT approximation algorithm, and the number 3 in fact comes from the sum of the coefficients $(2+1)$ in front of $\delta(y,X)$ and $\delta(y,B)$. An obvious way to try to improve this approximation factor is to reduce the coefficients in the terms $\delta(y,X)$ and $\delta(y,B)$ in the lemma for the case when δ is a Euclidean metric. Unfortunately, this straightforward attempt would not work: These constants are tight even for the line metric (see Figure 4.1).

We overcome this barrier via two ideas: **(i)** We allow a larger projection space, that is, instead of B, we project X on a larger set that is efficiently computable – the points in Y closest to the midpoints of pairs of points in B, therefore define the midpoint closure of $\mathsf{cl}(B) = B \cup \{\left(\pi_Y(\frac{b+b'}{2})\right) : b, b' \in B\}$, where $\pi_Y(p)$ represents the closest point in Y to point p; notice that this step crucially exploits the Euclidean space (so that the midpoints exist); and **(ii)** we "relax" the requirement and only aim to upper bound the projection distance among the "far" points in X. Given X and B, point $y \in Y$ is said to be ϵ-far if $\delta(y,X) \geq \epsilon\delta(y,B)$ (intuitively, these are the points that are sufficiently far from X compared to the distance from B). The rest of the points are called ϵ-near. We remark that each idea alone would not be sufficient to get an improved result.

We show that this new assignment property is enough to derive an improved FPT approximation for ROBUST (k,z)-CLUSTERING in Euclidean space. To achieve this, we find it necessary to separate the analysis into two distinct categories: ϵ-near and ϵ-far points. Initially, we observe that the approximation bound of [72] is tight for ϵ-near points. However, the same does not hold for ϵ-far points for which we can "save" some approximation factor. In the second part of our proof, we exploit the geometric inherent properties of the Euclidean space to provide an improved approximation factor specifically for the ϵ-far points. To do this, we partition our analysis into several individual cases. In each of the cases, we consider a special "region" of the space, determined by the relative position of a point p with respect to certain centers in the optimal solution O and in the bicriteria solution B. Using intricate geometric arguments of the Euclidean metric, we show the desired approximation factor for ϵ-far points. Finally, we demonstrate how a combined solution for ϵ-near and ϵ-far points achieves the improved approximation factor.

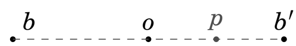

Figure 4.1. Here is an example showing that the algorithm of Goyal and Jaiswal [72] is tight even for line metric. Suppose $o \in O$ is an optimum facility that serves client p, and $b' \in B$ is a facility that serves client p in B. Suppose by our assignment rule we have $\sigma(o) = b$. We assume, w.l.o.g, that o is at the origin of the coordinate system and that $b = (-1,0)$. Now suppose client p is within distance $\frac{1}{2}$ from o and b', i.e., $p = (\frac{1}{2},0)$, and $b' = (1,0)$. In this example the cost for optimum solution is $\frac{1}{2}$, the cost for a solution B is $\frac{1}{2}$ and the cost for the algorithm is $\frac{3}{2}$, hence it shows that the algorithm is tight for the ratio and also for the connection costs.

4.4 EPAS for ROBUST k-MEDIAN (or SOCIALLY FAIR k-MEDIAN)

In this section, we will show an algorithm for ROBUST k-MEDIAN (i.e., ROBUST (k,z)-CLUSTERING with $z = 1$) in discrete finite Euclidean space. This algorithm is cleaner and essentially has all the ideas of the algorithm of Publication II for ROBUST (k,z)-CLUSTERING in doubling metrics.

Theorem 4.4.1 (EPAS for Euclidean metric of sub-logarithmic dimension)**.** *There is an algorithm that, given an instance of* ROBUST k-MEDIAN *in an Euclidean metric of d, computes a $(1+\epsilon)$-approximate solution for the instance, for every $\epsilon > 0$, in time $f(k,d,\epsilon)\mathsf{poly}(m,n)$, where $f(k,d,\epsilon) = \left(\left(\frac{2}{\epsilon}\right)^d k\log k\right)^{\mathcal{O}(k)}$.*

Note that the above algorithm runs in FPT time for $d = o(\log n)$. We also remark that the above result can be extended to the continuous $\mathbb{R}^d$. Throughout this section, we assume that the weight aspect ratio $\frac{\max_{p\in P} w(p)}{\min_{p'\in P} w(p')}$ and the distance aspect ratio $\frac{\max_{p,p'\in P}\delta(p,p')}{\min_{p\neq p'\in P}\delta(p,p')}$ are bounded by $\mathsf{poly}(n)$, some polynomial in n. We prove the theorem in two steps: first, in Section 4.4.1 we show an algorithm to obtain a coreset for the problem, and then, in Section 4.4.2 we show how to use this coreset to get the algorithm of Theorem 4.4.1.

4.4.1 Coreset for ROBUST k-MEDIAN

The idea for constructing a coreset for ROBUST k-MEDIAN (and ROBUST (k,z)-CLUSTERING) crucially relies on the following alternate but general definition.

Definition 4.4.1 (General Definition for ROBUST k-MEDIAN)**.** *Given a clustering metric space (P,F,δ) with $|P| = n$, a positive integer k, a collection of m group weight vectors $\mathcal{W} = \{\boldsymbol{w} : P \to \mathbb{R}^n_{\geq 0}\}$, find a k-element subset $X \subseteq F$ that minimizes $\max_{\boldsymbol{w}\in\mathcal{W}} \sum_{p\in P} \boldsymbol{w}_p \delta(p,X)$.*

Note that this definition generalizes Definition 4.1.1, by constructing the weight vector $\boldsymbol{w}^S$ for each group S such that $\boldsymbol{w}^S_p = 1$ if $p \in S$ and 0 otherwise. Further, note that Definition 4.4.1 allows a point to participate in different groups with different weights, which is not possible in Definition 4.1.1. We have the following coreset for ROBUST k-MEDIAN.

Theorem 4.4.2 (Coreset for ROBUST k-MEDIAN)**.** *Given an instance $\mathcal{I} = ((P,F,\delta),k,\mathcal{G})$ of* ROBUST k-MEDIAN *in Euclidean space of dimension d and $\epsilon > 0$, there is an algorithm that, in time $(2/\epsilon)^{\mathcal{O}(d)}\mathsf{poly}(n,m)$, computes another instance $\mathcal{I}' = ((P',F,\delta),k,\mathcal{W}')$ of* ROBUST k-MEDIAN *such that $P' \subseteq P : |P'| = (2/\epsilon)^{\mathcal{O}(d)} k\log n$, and for every $X \subseteq F, |X| = k$, it holds that*

$$(1-\epsilon)\mathsf{cost}(\mathcal{I},X) \leq \mathsf{cost}(\mathcal{I}',X) \leq (1+\epsilon)\mathsf{cost}(\mathcal{I},X).$$

We remark that the coreset $\mathcal{I}'$ obtained from this theorem has the same number of groups as in the original instance $\mathcal{I}$. We will show later that such a coreset is sufficient to get a parameterized approximation scheme with parameters k and d. Further, the exponential dependency on d on the point set size of the coreset is inevitable since ROBUST k-MEDIAN captures k-CENTER, for which such a lower bound is known [29, 17].

The Coreset Algorithm.

Algorithm 7: Coreset construction for ROBUST k-MEDIAN in $\mathbb{R}^d$

Data: Instance $\mathscr{I} = ((P, F, \delta), k, \mathscr{W})$ of ROBUST k-MEDIAN,
(α, β)-bicriteria solution B for $\mathscr{I}$

Result: Coreset $\mathscr{I}' = ((P', F, \delta), k, \mathscr{W}')$ for $\mathscr{I}$

Let $P' \leftarrow \emptyset$;
Let $\mathscr{W}'$ be collection of m weight vectors $\boldsymbol{w}' : P \to \mathbb{R}^n_{\geq 0}$ such that $\boldsymbol{w}' = \mathbf{0}$;
Let $\tau \leftarrow \max_{\boldsymbol{w} \in \mathscr{W}} ||\boldsymbol{w}||_1$;
Let $R = \mathsf{cost}(\mathscr{I}, B)/\alpha\tau$;
For each $b_i \in B$ for $j \in \{0, 1, \cdots, \lceil 2 \log \alpha n \Delta \rceil\}$, let $\mathscr{B}_i^j = \mathsf{ball}(b_i, 2^j R)$, and let $\mathscr{Q}_i^j = \mathscr{B}_i^j - \mathscr{B}_i^{j-1}$ with $\mathscr{Q}_i^0 = \mathscr{B}_i^0$;
Decompose each ball $\mathscr{B}_i^j$ into balls of radius each $\frac{\epsilon}{4\alpha} 2^j R$;
Associate each point $p \in P$ to a smallest ball containing p breaking ties arbitrary;
foreach $i \in [k]$ **do**
    foreach $j \in \{0, 1, \cdots, \lceil 2 \log \alpha n \Delta \rceil\}$ **do**
        foreach *smaller ball t of $\mathscr{B}_i^j$* **do**
            if $\exists p' \in t \cap Q_i^j$ **then**
                $P' \leftarrow P' \cup p'$;
                foreach $\boldsymbol{w} \in \mathscr{W}$ **do**
                    Set the corresponding weight vector $\boldsymbol{w}'[p] = \sum_{p' \in t \cap \mathscr{Q}_i^j} \boldsymbol{w}[p']$;
                end
                break;
            end
        end
    end
end
return $\mathscr{I}' := ((P', F, \delta), k, \mathscr{W}')$;

Algorithm 7 shows the detailed description for our algorithm for constructing the coreset for ROBUST k-MEDIAN. The high-level idea is as follows. The algorithm takes two inputs – an instance $\mathscr{I} = ((P, F, \delta), k, \mathscr{W})$ of ROBUST k-MEDIAN and an (α, β)-bicriteria solution B for $\mathscr{I}$, i.e., $|B| \leq \beta k$ and $\mathsf{cost}(\mathscr{I}, B) \leq \alpha \mathsf{OPT}$, where OPT denotes the optimal cost of $\mathscr{I}$. For our purpose, we assume that α, β are some fixed constants greater than 1. Let $R = \frac{\mathsf{cost}_{\mathscr{I}}(B)}{\alpha\tau}$, where $\tau := \max_{\boldsymbol{w} \in \mathscr{W}} ||\boldsymbol{w}||_1$. Let $\Delta = \frac{\max_{p \in P, \boldsymbol{w} \in \mathscr{W}} \boldsymbol{w}[p]}{\min_{p \in P, \boldsymbol{w} \in \mathscr{W}} \boldsymbol{w}[p]}$ be the weight aspect ratio of $\mathscr{I}$. Then, for each $b_i \in B$ and for $j \in \{0, 1, \ldots, \lceil 2\log(\alpha n \Delta) \rceil\}$, consider the balls $\mathscr{B}_i^j := \mathsf{ball}(b_i, 2^j R)$, and the corresponding rings $\mathscr{Q}_i^j = \mathscr{B}_i^j - \mathscr{B}_i^{j-1}$, for $j = \{1, \cdots, \lceil 2\log(\alpha n \Delta) \rceil\}$, with $\mathscr{Q}_i^0 = \mathscr{B}_i^0$. First note that every point $p \in P$ lies in some ball $\mathscr{B}_i^j$, since $\delta(p, B) \leq \frac{\mathsf{cost}_{\mathscr{I}}(B)}{\boldsymbol{w}_p} \leq \frac{\alpha\tau}{\boldsymbol{w}_p} R \leq R \alpha n \Delta$. Next, decompose every ball $\mathscr{B}_i^j$ into smaller balls each of radius $\frac{\epsilon}{4\alpha} R 2^j$. Since these balls can intersect, assume that every point $p \in P$ lies in exactly one smaller ball (for example, by associating p to the smallest ball containing p, breaking ties arbitrarily). Finally, for every ball $\mathscr{B}_i^j$ and every smaller ball t of $\mathscr{B}_i^j$ with $|t \cap \mathscr{Q}_i^j| \neq \emptyset$, pick an arbitrary point $p' \in t \cap \mathscr{Q}_i^j$ as the *representative* of (the points in) $t \cap \mathscr{Q}_i^j$, and add p' to the coreset P' with group weight vectors as follows. For every group vector $\boldsymbol{w} \in \mathscr{W}$, consider the corresponding weight vector $\boldsymbol{w}' \in \mathscr{W}'$, and set $\boldsymbol{w}'_{p'} := \sum_{p \in t \cap Q_i^j} \boldsymbol{w}_p$.

We have the following guarantee about this algorithm.

Lemma 4.4.1. *Algorithm 7 constructs a coreset* $\mathscr{I}' = ((P', F, \delta), k, \mathscr{W}')$ *of* $\mathscr{I}$ *such that* $|P'| = \mathscr{O}(\beta k \log(\alpha n) \left(\frac{\alpha}{\epsilon}\right)^{\mathscr{O}(d)})$ *in time* $\mathscr{O}(\beta k \log(\alpha n) \left(\frac{\alpha}{\epsilon}\right)^{\mathscr{O}(d)})\mathsf{poly}(n,m)$*, assuming* $\Delta = \mathsf{poly}(n,m)$*.*

Proof. The bound on $|P'|$ and the running time of Algorithm 7 follow from the parameters used in the algorithm. We now argue that $\mathscr{I}'$ is indeed a coreset for $\mathscr{I}$. Let OPT be the cost of optimal solution for $\mathscr{I}$. Fix a feasible solution $X \subseteq F, |X| = k$, and let $\hat{\boldsymbol{w}} \in \mathscr{W}$ be a maximizer of the ROBUST k-MEDIAN cost of $\mathscr{I}$ for X. We claim that, for any $\boldsymbol{w}' \in \mathscr{W}'$, it holds that

$$\mathsf{cost}(\boldsymbol{w}, X) - \epsilon \mathsf{cost}(\hat{\boldsymbol{w}}, X) \leq \mathsf{cost}(\boldsymbol{w}', X) \leq (1+\epsilon)\mathsf{cost}(\hat{\boldsymbol{w}}, X),$$

where $\boldsymbol{w} \in \mathscr{W}$ is the corresponding weight vector to $\boldsymbol{w}'$. To bound the error due to the coreset, we partition P into two parts – one that contains points close to B, i.e., points in $\mathscr{B}^0 := \cup_{i \in |B|} B_i^0$, and another part with points outside $\mathscr{B}^0$. For $p \in P$, let $p' \in P'$ be its representative in P'. First, consider the points in $\mathscr{B}^0$, and note that

$$|\sum_{p \in \mathscr{B}^0 \cap P} \boldsymbol{w}_p \delta(p, X) - \sum_{p' \in \mathscr{B}^0 \cap P'} \boldsymbol{w}'_{p'} \delta(p', X)| \leq \sum_{p \in \mathscr{B}^0 \cap P} \boldsymbol{w}_p \delta(p, p') \leq \frac{\epsilon R}{4\alpha} \sum_{p \in \mathscr{B}^0 \cap P} \boldsymbol{w}_p,$$

which is at most $\frac{\epsilon \mathsf{OPT}}{4}$, where we used $\delta(p, p') \leq \frac{\epsilon R}{4\alpha}$ for $p \in \mathscr{B}^0 \cap P$. On the other hand, for $p \in P$ outside $\mathscr{B}^0$, note that $2^{j-1} R \leq \delta(p, B) \leq 2^j R$, for some $j \geq 1$. Hence, we have $\delta(p, p') \leq \frac{\epsilon 2^j R}{4\alpha} \leq \frac{\epsilon \delta(p, B)}{2\alpha}$. Thus, we have

$$|\sum_{p \in P \setminus \mathscr{B}^0} \boldsymbol{w}_p \delta(p, X) - \sum_{p' \in P' \setminus \mathscr{B}^0} \boldsymbol{w}'_{p'} \delta(p', X)| \leq \sum_{p \in P \setminus \mathscr{B}^0} \boldsymbol{w}_p \delta(p, p') \leq \sum_{p \in P \setminus \mathscr{B}^0} \boldsymbol{w}_p \frac{2\epsilon \delta(p, B)}{4\alpha},$$

which is at most $\frac{\epsilon \mathsf{OPT}}{2}$. Now,

$$\begin{aligned}
\mathsf{cost}(\boldsymbol{w}', X) &\leq \mathsf{cost}(\boldsymbol{w}, X) + \epsilon \mathsf{OPT} \\
&\leq \mathsf{cost}(\boldsymbol{w}, X) + \epsilon \mathsf{cost}(\hat{\boldsymbol{w}}, X) \qquad \text{since } \mathsf{OPT} \leq \mathsf{cost}(\mathscr{I}, X) = \mathsf{cost}(\hat{\boldsymbol{w}}, X) \\
&\leq (1+\epsilon)\mathsf{cost}(\hat{\boldsymbol{w}}, X) \qquad \text{since } \mathsf{cost}(\boldsymbol{w}, X) \leq \mathsf{cost}(\hat{\boldsymbol{w}}, X).
\end{aligned}$$

Similarly, $\mathsf{cost}(\boldsymbol{w}', X) \geq \mathsf{cost}(\boldsymbol{w}, X) - \epsilon \mathsf{OPT} \geq \mathsf{cost}(\boldsymbol{w}, X) - \epsilon \mathsf{cost}(\hat{\boldsymbol{w}}, X)$ since $\mathsf{OPT} \leq \mathsf{cost}(\hat{\boldsymbol{w}}, X)$.

Now, we finish the proof as follows.

$$\mathsf{cost}(\mathscr{I}',X) = \max_{\boldsymbol{w}'\in\mathscr{W}} \mathsf{cost}(\boldsymbol{w}',X) \le (1+\epsilon)\mathsf{cost}(\hat{\boldsymbol{w}},X) = (1+\epsilon)\mathsf{cost}(\mathscr{I},X)$$

On the other hand,

$$\mathsf{cost}(\mathscr{I}',X) = \max_{\boldsymbol{w}'\in\mathscr{W}} \mathsf{cost}(\boldsymbol{w}',X) \ge \max_{\boldsymbol{w}\in\mathscr{W}} \mathsf{cost}(\boldsymbol{w},X) - \epsilon\mathsf{cost}(\hat{\boldsymbol{w}},X) = (1-\epsilon)\mathsf{cost}(\mathscr{I},X).$$

□

Proof of Theorem 4.4.2. We instantiate Algorithm 7 with the following bi-criteria solution for ROBUST k-MEDIAN due to [103]. The bounds follow since $\alpha = \mathcal{O}(1)$ and $\beta = \mathcal{O}(1)$.

Theorem 4.4.3 ([103])**.** *There exists a polynomial time algorithm for* ROBUST (k,z)-CLUSTERING *that, for every $\gamma \in (0,1)$, outputs at most $\frac{k}{1-\gamma}$ centers whose cost is bounded by $\frac{e^{O(z)}}{1-\gamma}$ times the optimal cost.*

□

4.4.2 EPAS for ROBUST k-MEDIAN

In this section, we describe the second algorithm that takes a coreset of an instance of ROBUST k-MEDIAN and produces a $(1+\epsilon)$-approximate solution. By scaling the distances in the instance of ROBUST k-MEDIAN we assume that the distances are between 1 and Δ', for some number Δ'. Towards this goal, first we show the following result based on Algorithm 8, when $\Delta' = \mathsf{poly}(n)$.

Theorem 4.4.4. *For any $\epsilon > 0$, Algorithm 8, on input $\mathscr{I} = ((P,F,\delta),k,\mathscr{W})$, computes $X \subseteq F : |X| \le k$ such that $\mathsf{cost}(\mathscr{I},X) \le (1+\epsilon)\mathsf{OPT}$ in time $(1/\epsilon)^{\mathcal{O}(dk)}(|P|\log n)^k \mathsf{poly}(n,m)$, where OPT denotes the optimal cost of $\mathscr{I}$.*

The idea of Algorithm 8 is based on the leader guessing approach of [52] that exploits the smaller size of P. Consider some optimal solution $O = (o_1,\cdots,o_k)$, and let $Q = (Q_1,\cdots,Q_k)$ be the corresponding clusters. Each cluster Q_i can be identified by a closest point ℓ_i to o_i in P. Let $\lambda_i := \delta(\ell_i,o_i)$. Then, note that in FPT time we can guess (arbitrary good approximation of) λ_i and ℓ_i for all $i \in [k]$ if $|P|$ is small and the distance aspect ratio is bounded polynomially. For correct guesses of (ℓ_i,λ_i), for all $i \in [k]$, consider the ball $B_i = \mathsf{ball}(\ell_i,\lambda_i)$, and note that $o_i \in B_i$. Now, observe that picking an arbitrary facility f from each B_i already gives a factor 3 approximation since any point p served by o_i in the optimal solution is now served by f, which is at distance at most $2\lambda_i \le 2\delta(p,o_i)$ from o_i. Decomposing every ball B_i into smaller balls yields a $(1+\epsilon)$-approximate solution.

Algorithm 8: $(1+\epsilon)$-approximation algorithm for ROBUST k-MEDIAN

Data: Instance $\mathscr{I} = ((P,F,\delta),k,\mathscr{W})$ of ROBUST k-MEDIAN
Result: $(1+\epsilon)$-approximate solution $X \subseteq F$

```
Let X ← ∅;
forall k-tuples (ℓ_1,⋯,ℓ_k) of P do
    forall k-tuples (λ_1,⋯,λ_k) radii of (ℓ_1,⋯,ℓ_k) that are power of (1+ε/10) do
        for i ∈ [k] do
            𝓑_i ← {ε/20-ball decomposition of ball(ℓ_i, λ_i)};
        end
        T_i ← {f ∈ F | f is an arbitrary facility in ball b ∈ 𝓑_i} ^a ;
        forall k-tuples (t_1,⋯,t_k) of T_1 × ⋯ × T_k do
            if cost(𝓘,{t_1,⋯,t_k}) < cost(𝓘,X) then
                X ← {t_1,⋯,t_k}
            end
        end
    end
end
return X;
```

[a] If $F = \mathbb{R}^d$ then $T_i \leftarrow \{x_b \in F \mid x_b \text{ is the center of ball } b \in \mathscr{B}_i\}$

Proof of Theorem 4.4.4.
First, we bound the runtime of the algorithm. The leader enumeration (first forall loop) requires at most $|P|^k$ loops, one for each k-tuple of P. Assuming $\Delta' = \mathsf{poly}(n)$, the radii enumeration (second forall loop) for the leaders requires at most $\left(\frac{\mathscr{O}(\log n/\epsilon)}{\epsilon}\right)^k = (\mathscr{O}(\log n/\epsilon))^k$ loops using discretized steps of size that is a power of $(1+\epsilon/10)$. Finally, there are at most $(1/\epsilon)^{\mathscr{O}(dk)}$ many k-tuples of $T_1 \times \cdots \times T_k$ since $|T_i| = (1/\epsilon)^{\mathscr{O}(d)}$, yielding the claimed runtime.

For correctness, we will show that, for any group $\boldsymbol{w} \in \mathscr{W}$, we have $\mathsf{cost}(\boldsymbol{w},X) \leq (1+\epsilon)\mathsf{cost}(\boldsymbol{w},O)$, which implies $\mathsf{cost}(\mathscr{I},X) \leq (1+\epsilon)\mathsf{cost}(\mathscr{I},O)$, where $O = \{o_1,\cdots,o_k\}$ is an optimal solution. Let Π_i be the set of points in P served by o_i, for $i \in [k]$. Let $\ell_i^* \in \Pi_i$ be a point in P that is closest to o_i. Let this distance be λ_i^*, i.e., $\lambda_i^* := \delta(\ell_i^*, o_i)$. We call ℓ_i^* as the leader of Π_i with radius λ_i^*. Let λ_i' be the smallest number equal to some power of $(1+\epsilon/10)$ that is larger than λ_i^*. Then, note that $\lambda_i' \geq \lambda_i^* \geq \frac{\lambda_i'}{(1+\epsilon/10)}$. Next, consider the $\frac{\epsilon}{20}$-ball decomposition of $\mathsf{ball}(\ell_i^*, \lambda_i')$, and let b_i^* be a ball containing o_i. Now, consider the iteration of the algorithm corresponding to leader-radii enumeration $(\ell_1^*,\ldots,\ell_k^*)$ and $(\lambda_1',\ldots,\lambda_k')$, and let $(t_1^*,\ldots,t_k^*)$ be the centers considered by the algorithm for the balls $(b_1^*,\ldots,b_k^*)$. Then, for any $p \in \Pi_i, i \in [k]$, we have that

$$\delta(p,t_i^*) \leq \delta(p,o_i) + \delta(o_i,t_i^*) \leq \delta(p,o_i) + \frac{\epsilon}{10}\lambda_i' \leq \delta(p,o_i) + \frac{\epsilon}{10}\left(1+\frac{\epsilon}{10}\right)\lambda_i^*$$

Therefore, $\delta(p,t_i^*) \le (1+\epsilon)\delta(p,o_i)$. Hence, for any group $\boldsymbol{w} \in \mathscr{W}$, we have $\mathsf{cost}(\boldsymbol{w},X) = \sum_{p\in P} \boldsymbol{w}_p \delta(p,X) \le (1+\epsilon)\mathsf{cost}(\boldsymbol{w},O)$. □

Proof of Theorem 4.4.1.
Given an instance $\mathscr{I} = ((P,F,\delta),k,\mathscr{W})$ of ROBUST k-MEDIAN and an accuracy parameter $\epsilon > 0$, we invoke Theorem 4.4.2 on $\mathscr{I}$ with parameter $\epsilon/10$ to obtain a coreset $\mathscr{I}' = ((P',F,\delta),k,\mathscr{W}')$ such that $P' \subseteq P : |P'| = (2/\epsilon)^{\mathscr{O}(d)} k \log n$. Then, we invoke Theorem 4.4.4 on $\mathscr{I}'$ with parameter $\epsilon/10$ to obtain $X \subseteq F : |X| \le k$ such that $\mathsf{cost}(\mathscr{I}',X) \le (1+\epsilon/10)\mathsf{OPT}(\mathscr{I}')$, where $\mathsf{OPT}(\mathscr{I}')$ is the optimal cost of $\mathscr{I}'$.

First, we analyze the overall running time. With $|P'| = (2/\epsilon)^{\mathscr{O}(d)} k \log n$, Theorem 4.4.4 runs in time $\left((2/\epsilon)^d k \log n\right)^{\mathscr{O}(k)} \mathsf{poly}(n,m)$, leading to $\left((2/\epsilon)^d k \log k\right)^{\mathscr{O}(k)} \mathsf{poly}(n,m)$ as the overall running time as desired. For correctness, consider

$$
\begin{aligned}
\mathsf{cost}(\mathscr{I},X) &\le (1+\epsilon/10)\mathsf{cost}(\mathscr{I}',X) && \text{by the coreset property}\\
&\le (1+\epsilon/10)^2\mathsf{OPT}(\mathscr{I}') && \text{by Algorithm 4.4.4}\\
&\le (1+\epsilon/10)^3\mathsf{OPT}(\mathscr{I}) && \text{by the coreset property}\\
&\le (1+\epsilon)\mathsf{OPT}(\mathscr{I}).
\end{aligned}
$$

□

4.5 Conclusion and Open Problems

In Publication II, we give a comprehensive understanding of the parameterized approximibility of ROBUST (k,z)-CLUSTERING in geometric spaces where the parameter is the number k of centers. We prove the following results: (i) For a universal constant $\eta_0 > 0.0006$, we devise a $3^z(1-\eta_0)$-factor FPT approximation algorithm for ROBUST (k,z)-CLUSTERING in discrete high-dimensional Euclidean spaces where the set of potential centers is finite. This shows that the lower bound of 3^z for general metrics [73] no longer holds when the metric has geometric structure. (ii) We show that ROBUST (k,z)-CLUSTERING in discrete Euclidean spaces is $(\sqrt{3/2}-o(1))$-hard to approximate for FPT algorithms, even if we consider the special case k-CENTER in logarithmic dimensions. This rules out an EPAS for the problem, giving a striking contrast with the EPAS of Publication I for the continuous setting where centers can be placed anywhere in the space. (iii) However, we obtain an EPAS for ROBUST (k,z)-CLUSTERING in discrete Euclidean spaces when the dimension is sublogarithmic (for the discrete problem, Publication I provides an EPAS only in dimension $o(\log\log n)$). Our EPAS works also for metrics of sub-logarithmic doubling dimension.

In the context of open problems in this research, there are two intriguing directions for exploration. The first avenue of interest is to achieve an approximation factor significantly better than 3^z for the problem. Second, on the flip side, it is also a compelling challenge to investigate whether there are stronger lower bounds for the FPT-approximation algorithms.

5. Clustering with Diversity Constraints

This chapter is based on Publication III [121] and studies another variant of fair clustering, called *diversity-aware clustering* [123], that ensures diversity amongst the selected clustering centers. Although most of our results extend to other classical clustering problems, we limit our attention to diversity-aware formulations of k-MEDIAN and k-MEANS.

5.1 Fair-Center Clustering

Consider the problem of selecting a committee of representatives for university students. In classical clustering problems, the goal in hand is to select this committee so as to minimize some clustering objective. For example, in the k-MEDIAN problem, we want to select a committee of size k such that the total (average) distance of the committee to university students is minimized. Here, the distance function, for example, can be the shortest path distance on the social network graph of the students. Although such (classical) formulations have been good enough for the past century, the idea of focusing only on minimizing the cost function (objective) fails to resonate with modern sentiments of equality, fairness, and inclusivity. For example, in our problem of selecting a committee, a student may not be comfortable talking to the committee if they feel the committee is not sensitive to their beliefs or background, even though they know people in the committee. On the contrary, having a committee that over-represents some *group* is also not ideal. Thus, in addition to minimizing the cost function, we would ideally want this committee to be as representative of the students as possible. Such a formulation that promotes diversity and inclusivity in the selected committee is known as *Diversity-aware clustering*. In this problem, apart from the client set C, the facility set F, and a metric d defined over them, we are additionally given a collection of subsets over facilities, called *groups*. The groups represent various profiles of the facility, such as expertise, gender, demographics, etc. Additionally, we are given a lower bound r_g requirement and an upper bound requirement R_g for each group g. Any committee $S \subseteq F$ of size k over facilities is said to *respect diversity-constraints* if S contains at most R_g and at least r_g members from each group g. Our task is to select a diversity-constraint respecting committee that minimizes a clustering objective. Thus, such committees avoid both under-representation and over-representation. Each member of the selected committee is called a *center*. Once we have selected the centers, there is a natural way to cluster the points in the client set: each client is assigned to the closest center. Hence, this clustering formulation corresponds to fair-center clustering.

5.1.1 Related Work

Even though diversity-aware formulations of the clustering problems are very recent [123], a limited variation [77], studied under the context of content distribution network application, has been known for a decade. This problem, known as RED-BLUE MEDIAN, is defined when we have two disjoint groups and each group has only an upper-bound requirement. The current best approximation factor in polynomial time is due to [68]. They show that the local search algorithm yields a $5+\epsilon$ approximation. A more general version of this problem, where the groups are disjoint, is called PARTITION-MEDIAN. This problem is a special version of the MATROID-MEDIAN problem [88], where the selected centers are required to form an independent set under the given matroid. The best approximation factor, due to [89], for this problem is $7.081+\epsilon$ in polynomial time. When allowed to use FPT time in k, [53] presents a $(2+\epsilon)$-approximation algorithm for this problem. We remark that, in our diversity-aware formulation, it is permissible for groups to intersect. This is essential because, in real-world scenarios, groups can exhibit a more complex nature (such as groups based on expertise sets), and hence PARTITION-MEDIAN is a restricted variant of our formulation. Unfortunately, PARTITION-MEDIAN failed to receive attention from the research community. On the other hand, formulation for avoiding under-representation is due to [123], who showed **NP**-hardness, fixed parameter intractability, and polynomial time inapproximibility to any multiplicative factor for the problem.

For classical problems such as k-MEDIAN, k-MEANS, and k-CENTER, extensive efforts have been made to understand their approximibility. While the best known approximation factor for k-MEDIAN in polynomial time stands at 2.675 due to [31], the best known lower bound is $1+2/e$ assuming $\mathbf{P} \neq \mathbf{NP}$ [75]. Recently, [53] presented a FPT-approximation algorithm for k-MEDIAN that achieved the $(1+2/e+\epsilon)$ factor. They also showed that this factor is essentially best for an FPT algorithm, assuming Gap-ETH, a stronger version of ETH.

5.1.2 Problem Definition

In this chapter, for ease of understanding, we will consider diversity-aware formulation for k-MEDIAN. Even though, in Publication III, we consider diversity-aware formulations for k-MEDIAN and k-MEANS, our algorithmic results actually hold for diversity-aware formulations for a more generalized clustering problem called (k,z)-CLUSTERING, where the objective is to minimize the sum of z^{th} power of distances. Further, as mentioned before, the original diversity-aware formulation [123] is defined only to avoid under-representations (each group has only a lower bound requirement). In the following, we provide a more generic definition of diversity-aware formulation that avoids both under-representation and over-representation. This definition captures diversity and inclusivity in the selected centers in a natural way. We would like to remark that our algorithms in Publication III work for this general formulation of the problem. Also, note that in this chapter, in order to be consistent with Publication III, we slightly change the notation of the underlying metric space.

Definition 5.1.1 (Diversity-aware k-MEDIAN (DIV-k-MED))**.** *Given a metric space (U,d), a set $C \subseteq U$ of clients, a set $F \subseteq U$ of facilities, an integer $0 \le k \le |F|$, and a collection, called groups, $\mathcal{G} = \{G_1, \cdots, G_t\}, G_i \subseteq F$ with integers $0 \le r_i \le R_i \le k$, the problem is find $S \subseteq F$, $|S| = k$ satisfying $r_i \le |S \cap G_i| \le R_i$ for all $i \in [t]$ such that the clustering cost of S,* $\mathrm{cost}(S) = \sum_{c \in C} d(c,S)$ *is minimized.*

An instance of DIV-k-MED is denoted as $I = ((U,d), F, C, \mathcal{G}, \vec{r}, \vec{R}, k)$, where $\vec{r}(i) := r_i$ and $\vec{R}(i) := R_i$. We denote by $r = \max_{i \in [t]} r_i$. Without loss of generality, we assume that the distance aspect ratio $\Delta := \frac{\max_{x \neq y} d(x,y)}{\min_{x \neq y} d(x,y)}$ is polynomially bounded in $|U|$ [53].

5.2 Thesis Contributions

5.2.1 Hardness Results

When parameterized by k, k-MEDIAN is known to be **W[2]**-hard [75], the corresponding diversity-aware formulation is also **W[2]**-hard. Note that the trivial algorithm that exhaustively searches for a feasible solution runs in time $\mathcal{O}^*\left(|F|^k\right)$. Unfortunately, the following proposition says that even for deciding whether or not the given instance of DIV-k-MED is feasible, this is the best algorithm to hope for; in fact, it also rules out any multiplicative factor approximation algorithm running slightly faster than the trivial algorithm. The proof of this proposition follows from the reduction of [123] from k-DOMSET and observing that any factor approximation algorithm for

DIV-k-MED in time $\mathcal{O}\left(|F|^{k-\epsilon}\right)$ solves the feasibility version of the problem and hence solves k-DOMSET, contradicting SETH.

Proposition 5.2.1. *Assume SETH. For all $k \geq 3$ and $\epsilon > 0$, there exists no $\mathcal{O}\left(|F|^{k-\epsilon}\right)$ algorithm to solve* DIV-k-MED *(*DIV-k-MEANS *resp.). Furthermore, there exists no $\mathcal{O}\left(|F|^{k-\epsilon}\right)$ algorithm to approximate* DIV-k-MED *(*DIV-k-MEANS *resp.) to any multiplicative factor.*

The tractability of DIV-k-MED gets even worse: the problem does not admit an FPT algorithm even when we are allowed to open $f(k)$ centres for any function f. This follows from the result of [116] that rules out finding a dominating set of size $f(k)$ for any f in FPT time.

Proposition 5.2.2. *For any function f, finding $f(k)$ facilities to approximate* DIV-k-MED *(*DIV-k-MEANS *resp.) cost to any multiplicative factor in FPT time with respect to the parameter k is* $\mathbf{W[1]}$*-hard.*

Furthermore, it may be promising to consider other natural parameters of the problem and design an FPT algorithm with these parameters. However, this approach is also futile. For example, having an FPT algorithm for the problem with parameters t and k implies an FPT algorithm for k-MEDIAN since DIV-k-MED captures k-MEDIAN when $t = 1$ (recall that t represents the number of groups). Similarly, adding more parameters such as the maximum lower bound $r = \max_{i \in [t]} r_i$ and the maximum number of groups to which a facility can belong $\mu := \max_{d \in F} |G_i \in \mathcal{G} : f \in G_i|$ are also unlikely to help since they are bounded from above by k and t, respectively.

Proposition 5.2.3. *Finding an optimal solution for* DIV-k-MED *(*DIV-k-MEANS *resp.) is* $\mathbf{W[2]}$*-hard with respect to parameters (k, r, μ, t).*

Given such a bleak outlook for the tractability of the problem, we relax our requirement further and ask if we can approximate the problem in FPT time by adding more parameters. We present a tight FPT-approximation algorithm parameterized by (k, t) and answer this question positively.

5.2.2 Algorithmic Results

Our main result in Publication III is the following FPT-approximation algorithm for DIV-k-MED with a tight approximation factor.

Theorem 5.2.1. *There exists a randomized $(1 + 2/e + \epsilon)$-approximation algorithm for* DIV-k-MED*, for $\epsilon > 0$, running in time $f(k, t, \epsilon) \cdot \mathrm{poly}(|U|)$, where $f(k, t, \epsilon) = \mathcal{O}\left(\left(\frac{2^t k^3 \log^2 k}{\epsilon^2 \log(1+\epsilon)}\right)^k\right)$. Further, assuming Gap-ETH the approximation ratio is tight for any FPT algorithm parameterized by (k, t).*

Recall that t denotes the number of groups. A surprising element of the above theorem is that, when allowed to use additional parameter t, it matches the approximation factor guarantee of k-MEDIAN, which is a considerably simpler version of our problem.

5.3 Key Ideas of the Algorithm

Our algorithm is based on the simple idea of finding feasible constraint patterns for the given instance and then finding an approximate solution for each feasible constraint pattern. First, we define what we mean by a feasible constraint pattern. Let $((U,d),F,C,\mathcal{G},\vec{r},\vec{R},k)$ be an instance of DIV-k-MED. For each group $f \in F$, let $\vec{\chi}_f \in \{0,1\}^t$ be the characteristic vector of f with respect to $\mathcal{G}$, i.e., for $i \in [t], \vec{\chi}_f(i) := 1$ if $f \in G_i$, otherwise 0. Next, for each $\vec{\gamma} \in \{0,1\}^t$, let $E(\vec{\gamma})$ denote all the facilities whose characteristic vector is $\vec{\gamma}$, i.e., $E(\vec{\gamma}) := \{f \in F \mid \vec{\chi}_f = \vec{\gamma}\}$. Finally, let $\mathcal{E} = \{E(\vec{\gamma}) \mid \vec{\gamma} \in \{0,1\}^t\}$. Note that $\mathcal{E}$ induces a partition on F. For $\vec{a},\vec{b} \in \{0,1\}^t$, we use the notation $\vec{a} \geq \vec{b}$ when $\vec{a}(i) \geq \vec{b}(i)$ for all $i \in [t]$.

Definition 5.3.1 (Feasible constraint pattern)**.** *Let $I = ((U,d),F,C,\mathcal{G},\vec{r},\vec{R},k)$ be an instance of* DIV-k-MED. *Let $W = \{E(\vec{\gamma_{i_1}}),\cdots,E(\vec{\gamma_{i_k}})\}$ be a k-multiset, where $E(\vec{\gamma_{i_j}}) \in \mathcal{E}$. Then, the constraint pattern associated with W is the vector $\vec{W}$ obtained by the element-wise sum of the characteristic vectors $\{\vec{\gamma_{i_1}},\cdots,\vec{\gamma_{i_1}}\}$. A constraint pattern $\vec{W}$ is said to be feasible for I if $\vec{r} \leq \vec{W} \leq \vec{R}$.*

The following lemma says that we can enumerate all feasible patterns in FPT time in tk. The proof is straightforward: for all k-multisets W of $\mathcal{E}$, we check that the corresponding constraint pattern $\vec{W}$ is feasible for I.

Lemma 5.3.1. *Given an instance $I = ((U,d),F,C,\mathcal{G},\vec{r},\vec{R},k)$ of the* DIV-k-MED *problem, we can enumerate all feasible constraint patterns for I in time $\mathcal{O}^*\left(2^{tk}\right)$.*

Note that once we have a feasible constraint pattern $W = \{E(\vec{\gamma_{i_1}}),\cdots,E(\vec{\gamma_{i_k}})\}$ for I, then choosing an arbitrary facility f_j from each $E(\vec{\gamma_{i_j}})$, for $j \in [k]$, forms a feasible solution $S_W \subseteq F$ to I. However, to obtain a solution with a good approximation factor, we will use ideas from [53]. In the following, we will show a slightly weaker result of obtaining a 3-approximation algorithm that illustrates the key ideas of the algorithm.

Theorem 5.3.1. *There exists a $(3+\epsilon)$-FPT approximation algorithm for $\epsilon > 0$ for* DIV-k-MED.

Proof sketch. Let $I = ((U,d),F,C,\mathcal{G},\vec{r},\vec{R},k)$ be a given instance of DIV-k-MED. Assume, without loss of generality, that I is feasible. Algorithm 9 presents the detailed description of our algorithm.

Algorithm 9: Algorithm for DIV-k-MED

Input: An instance $I = ((U,d), F, C, \mathscr{G}, \vec{r}, \vec{R}, k)$ of DIV-k-MED, error parameter $\epsilon > 0$

Output: A $(3+\epsilon)$-approximate solution T^* for I

foreach $\vec{\gamma} \in \{0,1\}^t$ **do**
  $E(\vec{\gamma}) \leftarrow \{f \in F : \vec{\gamma} = \vec{\chi}_f\}$;
end
$\mathscr{E} \leftarrow \{E(\vec{\gamma}) : \vec{\gamma} \in \{0,1\}^t\}$;
$C' \leftarrow$ CORESET$((U,d), F, C, k, \nu \leftarrow \epsilon/16)$;
$T^* \leftarrow \emptyset$;
foreach *multiset* $\{E(\vec{\gamma}_1), \cdots, E(\vec{\gamma}_k)\} \subseteq \mathscr{E}$ *of size* k **do**
  if $\vec{R} \geq \sum_{i \in [k]} \vec{\gamma}_i \geq \vec{r}$, *element-wise* **then**
    Duplicate facilities to make subsets in $\{E(\vec{\gamma}_1), \ldots, E(\vec{\gamma}_k)\}$ disjoint;
    $T \leftarrow k$-MED-k-PM$((U,d), \{E(\vec{\gamma}_1), \cdots, E(\vec{\gamma}_k)\}, C', \epsilon/4)$;
    `// Algorithm 10`
    if $cost(C', T) < cost(C', T^*)$ **then**
      $T^* \leftarrow T$;
    end
  end
end
return T^**;*

Algorithm 10: Algorithm for k-MED-k-PM

Input: An instance $J = ((U,d), \{E_1, \cdots, E_k\}, C')$ of k-MED-k-PM, error parameter $\epsilon' > 0$

Output: A $(3+\epsilon')$-approximate solution S^* for J

$S^* \leftarrow \emptyset$, $\eta \leftarrow e\epsilon'/2$;
foreach *ordered multiset* $\{c'_{i_1}, \cdots, c'_{i_k}\} \subseteq C'$ *of size* k **do**
  foreach *ordered multiset* $\Lambda = \{\lambda_{i_1}, \cdots, \lambda_{i_k}\}$ *such that* $\lambda_{i_j} \subseteq [[\Delta]_\eta]$ **do**
    for $j = 1$ *to* k **do**
      $S_j \leftarrow$ arbitrary facility in $\Pi_j := \{f \in E_j \mid d(c'_{i_j}, f) \leq \lambda_{i_j}\}$;
    end
    $S \leftarrow \cup_{j \in [k]} S_j$;
    if $cost(C', S) < cost(C', S^*)$ **then**
      $S^* \leftarrow S$;
    end
  end
end
return S^**;*

Analysis. Let $F^* = (f_1^*, \cdots, f_k^*)$ be an optimal set of facilities arbitrarily ordered. Let OPT denote the cost of F^*. Let $W^* = (E(\vec{\chi}_{f_1^*}), \cdots, E(\vec{\chi}_{f_k^*}))$. Observe that W^* is a feasible constraint pattern for I. Further, let $c_i^* \in C$ be the closest client (breaking ties arbitrarily) to f_i^*, and let $d(c_i^*, f_i^*) = \lambda_i^*$.

Let $\Pi_i^* = \{f \in E(\vec{\chi}_{f_i^*}) \mid d(f, c_i^*) = \lambda_i^*\}$. Now consider the solution $\hat{F} = (\hat{f}_1, \cdots, \hat{f}_k)$, where $\hat{f}_i$ is an arbitrary facility in Π_i^*. We claim that $\hat{F}$ is a 3-approximation to I.

Claim 5.3.1. $\mathsf{cost}(\hat{F}) \leq 3\mathsf{OPT}$.

Proof. First, note that $d(\hat{f}_i, f_i^*) \leq 2\lambda_i^*$. Consider $c \in C$, and suppose c is connected to f_i^* in F^*. Then, note that

$$d(c, \hat{F}) \leq d(c, \hat{f}_i) \leq d(c, f_i^*) + d(f_i^*, \hat{f}_i) \leq 3d(c, f_i^*),$$

since $\lambda_i^* \leq d(c, f_i^*)$. Hence, $\mathsf{cost}(\hat{F}) = \sum_{c \in C} d(c, \hat{F}) \leq 3\mathsf{OPT}$. □

Thus, it remains to show how to compute Π_i^*, c_i^*, and λ_i^*, for all $i \in [k]$ in $\mathsf{FPT}(k,t)$ time. We will show an enumeration procedure that enumerates all possible $(\lambda_1, \cdots, \lambda_k)$, $(c_1, \cdots, c_k)$ and $(\Pi_1, \cdots, \Pi_k)$ and construct a potential feasible solution $\tilde{F}$, and return $\tilde{F}$ with minimum cost. First, we will show how to compute c_i^* and λ_i^* in $\mathsf{FPT}(k)$ time. For $(\lambda_1^*, \cdots, \lambda_k^*)$, we use standard discretization procedure to discretize the distance in the multiples of $(1+\epsilon)$, for some $\epsilon > 0$. This discretization procedure enumerates all distances $(\lambda_1, \cdots, \lambda_k)$ and runs in $\mathsf{FPT}(k)$ time due to our assumption that the distance aspect ratio Δ is polynomially bounded in $|I|$. Now to compute $(c_1^*, \cdots, c_k^*)$, note that we can enumerate all k-tuples $(c_1, \cdots, c_k)$ in time $\mathcal{O}\left(|C|^k\right)$. To make this running time $\mathsf{FPT}(k)$, we use the fact that $|C|$ can be made $\mathcal{O}(k \log |C|)$ using coresets, by some loss in the cost function. Roughly speaking, C' is a coreset for C if the cost of every subset $S \subseteq F, |S| \leq k$ in C' is within a factor $(1 \pm \epsilon)$ of the cost of S in C. The following result provides us with the coreset of desired size for k-MEDIAN. However, note that the coreset for k-MEDIAN is also a coreset for DIV-k-MED.

Theorem 5.3.2 ([64], Theorem 4.9)**.** *Given a metric instance $I = ((U,d), C, F, k)$ of the k-*MEDIAN *problem, for each $\nu > 0$, $\delta < \frac{1}{2}$, there exists a randomized algorithm that, with probability at least $1-\delta$, computes a coreset $C' \subseteq C$ of size $|C'| = \mathcal{O}\left(\nu^{-2}(k \log |U| + \log \frac{1}{\delta})\right)$ in time $\mathcal{O}\left(k(|U|+k) + \log^2 \frac{1}{\delta} \log^2 |U|\right)$.*

Finally, we will show how to compute $(\Pi_1^*, \cdots, \Pi_k^*)$ in $\mathsf{FPT}(k,t)$ time. Recalling Lemma 5.3.1, we can enumerate all feasible constraint patterns for I over $\mathscr{E}$ in time $\mathcal{O}^*\left(2^{tk}\right)$. Hence, during enumeration, some iteration will correspond to the k-multiset $W^* = (E(\vec{\chi}_{f_1^*}), \cdots, E(\vec{\chi}_{f_k^*}))$, whose constraint pattern is a feasible constraint pattern due to F^*. This, together with the above enumeration for $(\lambda_1, \cdots, \lambda_k)$ and $(c_1, \cdots, c_k)$ allows us to compute $(\Pi_1, \cdots, \Pi_k)$ in $\mathsf{FPT}(k,t)$ time. For each possible combination of enumerations, we compute a potential feasible solution $\tilde{F}$, and we return $\tilde{F}$ with the minimum cost. Note that the errors due to discretization and coresets result into a loss of $(1+\epsilon)$ factor in the approximation, yielding $(3+\epsilon)$-approximate solution. □

5.4 Conclusion and Open Problems

In Publication III, we provide a complete characterization of the parameterized complexity of DIV-k-MED (and DIV-k-MEANS). In particular, we showed an FPT-approximation algorithm with a tight approximation factor for DIV-k-MED (and DIV-k-MEANS). Furthermore, this approximation factor matches that of k-MEDIAN, which is a special case of DIV-k-MED.

It is worth noting that the FPT-inapproximibility results for DIV-k-MED primarily arise from the challenge of finding a feasible solution (or finding a diversity-constraints respecting solution), which essentially boils down to solving the HITTING SET problem. Consequently, an interesting direction of exploration lies in investigating the polynomial time approximibility of DIV-k-MED for instances where finding a feasible solution is relatively straightforward. For example, consider cases where the groups are disjoint. In such scenarios, finding a feasible solution is easy. This particular setting can be viewed as a special case of MATROID-MEDIAN, where the underlying matroid is derived from the groups' partition. Notably, this setting inherits the polynomial time $(7.081+\epsilon)$-approximation factor of MATROID-MEDIAN [89]. Consequently, a pertinent question arises: Is this factor the best achievable approximation ratio for DIV-k-MED with disjoint groups? Additionally, exploring inapproximability results in this setting would present an interesting research problem. Finally, studying the problem under specific metrics such as bounded treewidth, planar metrics, and other graph metrics opens up additional avenues of investigation and is likely to yield valuable insights.

Part II

Lower Bounds for Graph Packing Problems

6. Parameterized Complexity of Compact Set Packing

This chapter is based on Publication IV [69] and studies the parameterized complexity of a fundamental **NP**-hard problem called SET PACKING in a "compact" regime. The SET PACKING problem is given a set system $(\mathscr{U},\mathscr{S})$ and a positive integer r to find r sets in $\mathscr{S}$ that are pairwise disjoint. That is, we want to find a packing of r sets in $\mathscr{S}$. This problem is also known as the HYPERGRAPH MATCHING problem in graph theory. To see the equivalence, note that we can view the elements of $\mathscr{U}$ as the vertices of a hypergraph H with sets in $\mathscr{S}$ as its hyperedges. Thus, the problem of finding an r-packing in $(\mathscr{U},\mathscr{S})$ is the same as finding a matching of size r in H. This problem is one of the fundamental problems in combinatorial optimization with numerous applications [127]. While this problem captures many classical combinatorial problems such as maximum independent set (or maximum clique) [9], k-dimensional matching, and other graph packing problems [44, 80], this generalization also makes the problem intractable in several regimes. For instance, the trivial algorithm that enumerates all r-sized subsets of $\mathscr{S}$ running in time $\mathscr{O}^*(|\mathscr{S}|^r)$ is the (asymptotically) fastest algorithm to hope for, unless *Exponential Time Hypothesis (ETH)* fails. Similarly, even finding non-trivial approximations to the problem is notoriously hard [81, 78]. To combat these negative results, several restrictions on the problem have been studied. For example, in the k-SET PACKING problem, we have the additional restriction that the sets in $\mathscr{S}$ are bounded in size by k. This problem has received a great deal of attention from the research community. However, it turns out that k-SET PACKING captures the independent set problem in bounded degree graphs, which again is a notoriously hard problem to approximate [10, 19]. On the algorithmic front, the best known approximation for this problem is $(k+1+\epsilon)/3$ [42, 59], while the lower bound has been stuck at $\Omega(k/\ln k)$ [82] for almost two decades now.

Another regime where SET PACKING has been studied extensively is the parameterized complexity regime. Here, the problem is known as PARAMETERIZED SET PACKING (PSP) with parameter r. Unfortunately, PSP remains intractable and is, in fact, **W[1]**-complete [63]. Even from the parameterized approximations perspective, the problem does not admit any non-trivial factor [36], assuming *Gap-ETH*, a stronger version of ETH. Very recently, assuming a weaker hypothesis of $\mathsf{FPT} \neq \mathbf{W[1]}$, [114] ruled out an FPT algorithm that can find a packing of size $r/r^{1/H(r)}$ for any increasing function $H(\cdot)$, when given a promise that there is an r-packing in the instance. Thus, SET PACKING is a notoriously hard problem in almost all computational regimes, even with restrictions.

In this chapter, in the quest for tractable instances, we consider yet another restriction of SET PACKING, which we call the *compact* regime. Here, we study the computational complexity of PSP on compact instances – instances that have a small universe. More precisely, in the COMPACT PSP problem, we have that $|\mathscr{U}| = f(r)\cdot\Theta(\mathsf{poly}(\log|\mathscr{S}|))$, for some function $f(r) \geq r$. Apart from algorithmic motivation, compact instances have recently been used as an intermediate step to show FPT-inapproximibility of (non-compact) classical problems (see, e.g., [25, 99] where the compact instances were used in proving FPT-inapproximability of the k-EvenSet and Dominating Set). Recently, [101] showed that the compact version of SET r-COVERING is $\mathbf{W[1]}$-complete, while it is known that the corresponding non-compact problem is $\mathbf{W[2]}$-complete. We hope that studying COMPACT PSP would lead to some ideas that would be useful in proving tight FPT inapproximability of PSP (that is, to weaken the Gap-ETH assumption used in [36]). We would like to remark on the choice of defining compact instance on a small universe size and not on a small $|\mathscr{S}|$. Note that, when $|\mathscr{S}| = f(r)\cdot\Theta(\mathsf{poly}(\log|\mathscr{U}|))$, the enumeration algorithm running in time $\mathcal{O}^*(|\mathscr{S}|^r)$ already yields an FPT algorithm [33].

6.1 Compact Regime

Definition 6.1.1 (PARAMETERIZED SET PACKING (PSP))**.** *Given a collection of sets $\mathscr{S} = \{S_1,\ldots,S_m\}$ over an universe $\mathscr{U} = \{e_1,\ldots,e_n\}$, an integer r which is also the parameter, the* PSP *problem asks if there is a collection of sets $\mathscr{S}' \subseteq \mathscr{S}$ such that $|\mathscr{S}'| = r$ and $S_i \cap S_j = \emptyset$ for every $S_i \neq S_j \in \mathscr{S}'$. An instance of* PSP *is denoted as $(\mathscr{U},\mathscr{S},r)$.*

COMPACT PSP is defined when the instances have $|\mathscr{U}| = f(r)\cdot\mathsf{poly}(\log|\mathscr{S}|)$, for some function $f(r) \geq r$.

6.2 Existing Works

In this section, we compare our contribution with existing works to highlight its significance. To our best knowledge, the compact version of combinatorial problems has not previously been formalized and investigated. However, several existing reductions already imply the hardness of a compact version of some of the combinatorial problems. Here we review and compare the related results.

Our Contribution. As far as we know, there are no results showing **W[1]**-hardness of COMPACT PSP, and hence the corresponding dichotomy (Theorem 6.3.1). The key contribution of Publication IV is to show the hardness result for COMPACT PSP. On the way, we also show an ETH-based almost tight running time lower bound for COMPACT PSP, with tight (up-to $\log r$ factor) universe size $|\mathscr{U}| = r \cdot \Theta(\log|\mathscr{S}|)$. Interestingly, we show both of these results with a single FPT reduction. In addition, we extend our framework to improve the running time lower bounds for COMPACT r-VECTORSUM.

Next we survey some known hardness results for SET r-COVERING in the compact regime and argue their limitations in extending them to PSP. In particular, **[85, Lemma 25]** shows a reduction from SGI to a variant of SET r-COVERING called COMPACT EXACT r-COVERING, where we want to find an r-packing that is also a covering (in fact, they show hardness for COMPACT SET r-COVERING. But a closer inspection of their construction shows that the intended set cover is also a packing). The high-level idea of the construction is similar to ours: first assign each vertex of G a logarithm-length binary pattern vector. Then, create two kinds of sets: V-sets that capture the mapping of the vertices and E-sets that capture the mapping of edges. The idea is to use the pattern vectors to create these sets so that there is an isomorphic copy of H in G if and only if there are $|V_H|$ many V-sets and $|E_H|$ many E-sets covering the universe exactly once. However, if we consider the soundness (No case) proof of this reduction, then it crucially relies on the fact that no candidate solution can cover the entire universe exactly once. In fact, it is quite easy to find r sets that are mutually disjoint but do not form a cover. Therefore, it fails to yield hardness for COMPACT PSP.

The heart of our construction lies in ensuring that, in the No case, any r sets intersect. To this end, we construct a combinatorial gadget called the Compatible Intersecting Set System (ISS) pair. This gadget is a pair of set systems $(\mathscr{A}, \mathscr{B})$ over a universe U that guarantees two properties: First, every pair of sets within each set system intersects, and second, for any set $a \in \mathscr{A}$, there exists $b \in \mathscr{B}$ such that a intersects every set in $\mathscr{B}$ except b. Further, we present a simple greedy algorithm that finds such compatible ISS pairs $(\mathscr{A}, \mathscr{B})$ over a universe of size N, each having roughly $2^{\Omega(N)}$ sets. Note that this gadget, which we use to build our compact hard instance, also has a "compact" universe. While, on the other hand, **[115]** shows COMPACT SET r-COVERING is **W[1]**-hard using a reduction from k-CLIQUE to SET r-COVERING with $r = \Theta(k^2)$ and $|\mathscr{U}| = r^{3/2} \cdot \Theta(\log|\mathscr{S}|)$, but does not yield a tight ETH-based running time lower bound. In contrast,[111] shows such a tight ETH lower bound for COMPACT SET r-COVERING: requiring time $|\mathscr{S}|^{\Omega(r)}$, which can be easily modified to obtain a similar running time lower bound for COMPACT EXACT r-COVERING (by reducing from

1-IN-3-SAT, instead from 3-SAT).

6.3 Thesis Contribution

The main result is the following dichotomy of PARAMETERIZED SET PACKING.

Theorem 6.3.1 (Dichotomy)**.** *The following dichotomy holds for* PSP.

- *If* $|\mathscr{U}| = f(r) \cdot o(\log|\mathscr{S}|)$*, for any* f*, then* PSP *is in FPT.*
- PSP *remains* **W[1]**-*hard even when* $|\mathscr{U}| = r \cdot \Theta(\log|\mathscr{S}|)$.

The algorithmic result follows from well-known dynamic programming based algorithms [26, 60] that run in time $O^*(2^{|\mathscr{U}|})$, which is FPT time when $|\mathscr{U}| = f(r)o(\log|\mathscr{S}|)$. The main contribution of this work is the **W[1]**-hardness of PSP even when $|\mathscr{U}| = r \cdot \Theta(\log|\mathscr{S}|)$. To this end, we show an FPT-reduction from SUBGRAPH ISOMORPHISM (SGI) to COMPACT PSP. The hardness result follows since SGI is **W[1]**-hard. In fact, our hardness result can be strengthened by assuming the Exponential Time Hypothesis (ETH) [60] to obtain the following result.

Theorem 6.3.2. COMPACT PSP *requires time* $|\mathscr{S}|^{\Omega(r/\log r)}$ *even when* $|\mathscr{U}| = r \cdot \Theta(\log|\mathscr{S}|)$*, unless ETH fails.*

The result of Theorem 6.3.2 follows from the ETH-hardness result of SGI due to [104] and from the fact that the hardness reduction of Theorem 6.3.1 is parameter preserving up to a multiplicative constant. Note that since PSP can be trivially solved by enumeration in time $O^*(|\mathscr{S}|^r)$, the above result says that, even for the compact instances, this is essentially our best hope, up to a log factor in the exponent. An interesting consequence of the dichotomy theorem coupled with Theorem 6.3.2 is the fact that, as soon as instances get asymptotically smaller, not only do we beat the enumerative algorithm, but we actually obtain an FPT algorithm. We would like to remark that the universe size in Theorem 6.3.2 is tight (up-to $\log r$ factor) since having $|\mathscr{U}| = o(r/\log r) \cdot \Theta(\log|\mathscr{S}|)$ would already allow $|\mathscr{S}|^{o(r/\log r)}$ time algorithm. Further, note that for **W[1]**-hardness, it is sufficient to have $|\mathscr{U}| = f(r) \cdot \Theta(\log|\mathscr{S}|)$, for some f, since we can add $f(r) - r$ new sets each with a unique dummy element and inflate the parameter to $f(r)$. However, this is not true for ETH-based running time lower bounds, as such inflation fails to transfer the lower bounds asymptotically. Finally, we would like to point out that after our article was made public, Huairui Chu [48] improved the lower bound of Theorem 6.3.2 to rule out $|\mathscr{S}|^{\Omega(r)}$ time for COMPACT PSP.

Further, we extend the construction of our framework (Theorem 6.3.3) to improve the running time lower bound (matching the trivial upper bound up to a log factor in the exponent) for the compact version of r-VECTORSUM: Given a collection $\mathscr{C}$ of N vectors in $\mathbb{F}_2^d$ and a target vector $\vec{b} \in \mathbb{F}_2^d$, r-VECTORSUM asks if there are r vectors in $\mathscr{C}$ that sum to $\vec{b}$. COMPACT r-VECTORSUM is defined when $d = f(r) \cdot \mathsf{poly}(\log N)$, for some $f(r) \geq r$. The present bound of [25] rules out $N^{o(\sqrt{r})}$ time for COMPACT r-VECTORSUM under ETH.

Theorem 6.3.3. COMPACT r-VECTORSUM *requires time* $N^{\Omega(r/\log r)}$*, even when* $d = r \cdot \Theta(\log N)$*, unless ETH fails.*

6.4 Notations

For a graph $G = (V,E)$ and a vertex $v \in V$, denote by $N(v)$, the set of vertices adjacent to v. For a finite universe U and $s \subseteq U$, denote by $\bar{s}$ as the complement of s under U, i.e., $\bar{s} := U \setminus s$. Similarly, for a family of sets $S = \{s_1, \cdots, s_M\}$ over U, we denote by $\mathsf{comp}(S) = \{\bar{s}_1, \cdots, \bar{s}_M\}$. Further, for a subset $s \subseteq U$ and a sub-universe $U' \subseteq U$, denote by $s\mid_{U'}$ the restriction of s on sub-universe U', i.e., $s\mid_{U'} := s \cap U'$. Similarly, for a family of sets $S = \{s_1, \cdots, s_M\}$ over U, denote by $S\mid_{U'}$ as the restriction of every set of S on U', i.e., $S\mid_{U'} := \{s_1\mid_{U'}, \cdots, s_M\mid_{U'}\}$. For a set system $A = (U_A, S_A)$, we denote the complement set system by $\bar{A} = (U_A, \mathsf{comp}(S_A))$. For $s, t \subseteq U$, we say s and t intersect if $s \cap t \neq \emptyset$.

6.5 Key Techniques

In this section, we show an alternate and simplified proof of Theorem 6.3.1 that highlights all the essential ideas of the proof in Publication IV. Note, however, that this proof yields a slightly weaker lower bound under ETH, and hence a weaker result than Theorem 6.3.2. First, let us define an alternate but equivalent definition of k-CLIQUE.

Definition 6.5.1 (k-CLIQUE)**.** *Given a positive integer k and two graphs* $G = (V,E)$ *and* $H = (W,F)$*, where H is a complete graph on k vertices and parameter k, the* k-CLIQUE *problem asks if there is an isomorphic copy of H in G.*

We will show the following reduction that implies the hardness result of Theorem 6.3.1 since the reduction produces an instance of COMPACT PSP.

Theorem 6.5.1. *There is an FPT-reduction that, for every instance* $\mathscr{I} = (G = (V,E), k)$ *of* k-CLIQUE *with* $|V_G| = n$ *and* $|E_G| = m$*, computes an instance* $\mathscr{J} = (\mathscr{U}, \mathscr{S}, r)$ *of* PSP *with* $|\mathscr{U}| = \Theta(k \log n)$*,* $|\mathscr{S}| = nk + m\binom{k}{2}$*, and* $r = k + \binom{k}{2}$ *such that G has a* k-CLIQUE *if and only if* $\mathscr{J}$ *has a packing of size r.*

To prove the above theorem, we first define the gadget used in this reduction.

6.5.1 Compatible Intersecting Set System Pair

A set system $\mathscr{A} = (U_A, S_A)$ is called an (M,N)-intersecting set system (ISS) if it contains M sets over N elements such that every pair $s,t \in S_A$ intersects.

Definition 6.5.2 (Compatible ISS pair)**.** *Given two ISS $\mathscr{A} = (U, S_A)$ and $\mathscr{B} = (U, S_B)$ on a universe U, we say that $(\mathscr{A}, \mathscr{B})$ is a compatible ISS pair if there exists an efficiently computable bijection $f : S_A \to S_B$ such that*

- *(Complement partition) $\forall s \in S_A$, s and $f(s)$ form a partition of U, and*
- *(Complement exchange) $\forall s \in S_A$, $\mathscr{A}_s := (U, (S_A \setminus \{s\}) \cup \{f(s)\})$ is an ISS.*

Since f is a bijection, we have $|S_A| = |S_B|$, and $\forall t \in S_B$, the set system $\mathscr{B}_t := (U, (S_B \setminus \{t\}) \cup \{f^{-1}(t)\})$ is also an ISS. Also, for $(s,t) \in (S_A, S_B)$, if $s \cup t = U$, then $t = f(s)$.

The following lemma, whose proof is present in Publication IV, efficiently constructs a compatible (M,N)-ISS pair that has a small universe, which is the key to our hardness proof.

Lemma 6.5.1. *For even $N \geq 2$, we can compute a compatible (M,N)-ISS pair $(\mathscr{A}, \mathscr{B})$ with $M \geq 2^{N/2-1}$ in time polynomial in M and N. Further, $\mathscr{B} = \overline{\mathscr{A}}$.*

6.5.2 Proof of Theorem 6.5.1.

Let $V = \{v_1, \cdots, v_n\}$, and we fix an ordering of vertices of V. For simplicity, we fix the natural ordering, i.e., $v_i < v_j$ if $i < j$. Further, assume vertices in an edge are ordered in the same way, i.e., $E \ni e = (v_i, v_j)$ if $i < j$. Similarly, let $W = \{w_1, \cdots, w_k\}$, and we again use the natural ordering on the vertices of W. Let $(\mathscr{A}, \overline{\mathscr{A}})$ be the compatible (M,N)-ISS pair given by Lemma 6.5.1, for $N = 2\lceil \log(n+1) \rceil + 2$. We call $\mathscr{A} = (U_A, S_A)$ and $\overline{\mathscr{A}} = (U_A, \mathsf{comp}(S_A))$ as the *base ISS gadgets*. Let $S_A = \{s_1, \cdots, s_M\}$. Since $M \geq 2^{N/2-1} > n$, every $v_i \in V$ can be identified by the set $s_i \in S_A$ corresponding to the index $i \in [M]$. We will use $k(k-1)$ copies of $(\mathscr{A}, \overline{\mathscr{A}})$ to create an instance $\mathscr{J} = (\mathscr{U}, \mathscr{S}, r)$ of PSP as follows. For each $w_j \in W$ and $w_{j'} \in N(w_j)$, we create a copy of $(\mathscr{A}^{(j,j')}, \mathscr{S}^{(j,j')})$ of $(\mathscr{A}, \overline{\mathscr{A}})$. Note that we create two distinct copies – $(\mathscr{A}^{(j,j')}, \mathscr{S}^{(j,j')})$ and $(\mathscr{A}^{(j',j)}, \mathscr{S}^{(j',j)})$, one for each $w_j, w_{j'} \in W$. Further, for each $j \in [k]$, we define $U_A^j = \bigcup_{w_{j'} \in N(w_j)} U_A^{(j,j')}$. The universe $\mathscr{U}$ of $\mathscr{J}$ is the union of all the universes U_A^j, i.e.,

$$\mathscr{U} := \bigcup_{j=1}^{k} U_A^j.$$

The sets in $\mathscr{S}$ are of two types: V-sets and E-sets, as defined below.
V-sets: For $v_i \in V$ and $w_j \in [k]$, we create a set $S_{v_i \leftrightarrow w_j}$ as follows.

$$S_{v_i \leftrightarrow w_j} := \bigcup_{w_{j'} \in N(w_j)} s_i^{(j,j')}$$

E-sets: For each edge $(v_i, v_{i'}) \in E$ and for $(w_j, w_{j'}) \in F$, we create a set $S_{(v_i,v_{i'}) \leftrightarrow (w_j,w_{j'})}$ as follows.

$$S_{(v_i,v_{i'}) \leftrightarrow (w_j,w_{j'})} := \bar{s}_i^{(j,j')} \cup \bar{s}_{i'}^{(j',j)}$$

Parameter: Set $r := k + \binom{k}{2}$.

This concludes the construction. Before we prove its correctness, we note the size of the constructed instance, $\mathscr{J}$. First, $r = \Theta(k^2)$. Then, $|\mathscr{U}| = \sum_{i=1}^{k} |U_A^j| = \sum_{i=1}^{k}(k-1)N = \Theta(k \log n)$, and $|\mathscr{S}| = nk + m\binom{k}{2}$.

Yes case. Suppose there is a k-CLIQUE $Q = \{v_{i_1}, \ldots, v_{i_k}\} \subseteq V$ in G, where the vertices of Q are in the natural order. Then, consider the following collection of V-sets and E-sets: $T_V := \bigcup_{j \in [k]} S_{v_{i_j} \leftrightarrow w_j}$ and $T_E := \bigcup_{(v_{i_j}, v_{i_{j'}}) \in E} S_{(v_{i_j}, v_{i_{j'}}) \leftrightarrow (w_j, w_{j'})}$, and let $T = T_V \cup T_E$. Note that these sets exist due to our construction and the existence of Q. Further, $|T| = |T_V| + |T_E| = k + \binom{k}{2}$, as required. Now, we claim that T forms a packing in $\mathscr{J}$. Towards this goal, note that it is sufficient to show that $T|_{U_A^j}$ forms a packing for all $w_j \in W$. Further, it is sufficient to show that $T|_{U_A^{(j,j')}}$ forms a packing for $w_j \in W$ and $w_{j'} \in N(w_j)$. However, this is true since

$$T|_{U_A^{(j,j')}} = \{T_V|_{U_A^{(j,j')}}, T_E|_{U_A^{(j,j')}}\} = \{s_{i_j}^{(j,j')}, \bar{s}_{i_j}^{(j,j')}\}.$$

No case. Suppose there is an r-packing $T \subseteq S$ in $\mathscr{J}$. Let $T_V \subseteq T$ and $T_E \subseteq T$ be the V-sets and E-sets of T. Note that our construction lets us identify V-sets and E-sets in T. First, observe that $|T_V| = k$ and $|T_E| = \binom{k}{2}$. This is because sets in T_V form a packing, and each set in T_V blocks $(k-1)N$ elements of U_A^j, for some $j \in [k]$, due to the intersecting property. Since $|\mathscr{U}| = k(k-1)N$, it follows that $|T_V| \leq k$. Similarly, sets in T_E form a packing, and each set in T_E blocks $2N$ elements of $\mathscr{U}$ due to the intersecting property. Hence, $|T_E| \leq \binom{k}{2}$. Now since $|T_V| + |T_E| = k + \binom{k}{2}$, it must be that $|T_V| = k$ and $|T_E| = \binom{k}{2}$. The following lemma is crucial for this case.

Lemma 6.5.2 (Covering Lemma). *T covers $\mathscr{U}$.*

Proof. Note that each set in T_V contains $(k-1)N/2$ elements of $\mathscr{U}$, and each set in T_E contains N elements of $\mathscr{U}$. Hence, T_V contains $k(k-1)N/2$ elements of $\mathscr{U}$, and T_E contains $\binom{k}{2}N$ elements of $\mathscr{U}$ because $|T_V| = k$ and $|T_E| = \binom{k}{2}$. Since T is a packing, it holds that T covers exactly $k(k-1)N = |\mathscr{U}|$ elements of $\mathscr{U}$. □

Next, we need the following claim that relies on the above covering lemma.

Claim 6.5.1. *For $1 \le j < j' \le k$, if $S_{v_i \leftrightarrow w_j}, S_{v_{i'} \leftrightarrow w_{j'}} \in T_V$, then $S_{(v_i,v_{i'}) \leftrightarrow (w_j,w_{j'})} \in T_E$.*

Proof. Suppose $S_{v_i \leftrightarrow w_j}, S_{v_{i'} \leftrightarrow w_{j'}} \in T_V$, then note that $T_E \mid_{U_A^{(j,j')}} = \bar{s}_i^{(j,j')}$ and $T_E \mid_{U_A^{(j',j)}} = \bar{s}_{i'}^{(j',j)}$, since T is a packing. Using the covering lemma, it must be that $S_{(v_i,v_\ell) \leftrightarrow (w_j,w_{j'})} \in T_E$ for some $v_\ell \neq v_i$. But this means that $\bar{s}_{i'}^{(j',j)} = T_E \mid_{U_A^{(j',j)}} = S_{(v_i,v_\ell) \leftrightarrow (w_j,w_{j'})} \mid_{U_A^{(j',j)}} = \bar{s}_\ell^{(j',j)}$. This implies that $v_\ell = v_{i'}$, as desired. □

A straight-forward application of this claim and the covering lemma helps us prove the following lemma, which asserts that each vertex of G has at most one V-set in T_V, and each edge of G has at most one E-set in T_E.

Lemma 6.5.3. *For each $v_i \in V$, there is at most one V-set $S_{v_i \leftrightarrow w_j}$ in T_V, for some $w_j \in W$. Similarly, for each edge $(v_i, v_{i'}) \in E$, there is at most one E-set $S_{(v_i,v_{i'}) \leftrightarrow (w_j,w_{j'})}$ in T_E, for $j \neq j' \in [k]$.*

Proof. Suppose $S_{v_i \leftrightarrow w_j}, S_{v_i \leftrightarrow w_{j'}} \in T_V$, for some $v_i \in V$, and $1 \le j < j' \le k$. Then, using Claim 6.5.1, we have that $S_{(v_i,v_i) \leftrightarrow (w_j,w_{j'})} \in T_E$, implying $(v_i, v_i) \in E$, a contradiction. Similarly, for some edge $(v_i, v_{i'}) \in E$, suppose $S_{(v_i,v_{i'}) \leftrightarrow (w_j,w_{j'})}, S_{(v_i,v_{i'}) \leftrightarrow (w_\ell,w_{\ell'})} \in T_E$, for some $(j, j') \neq (\ell, \ell')$. Without loss of generality, assume $j \neq \ell$. Then, since $T_E \mid_{U_A^{(j,j')}} = \bar{s}_i^{(j,j')}$ and $T_E \mid_{U_A^{(\ell,\ell')}} = \bar{s}_i^{(\ell,\ell')}$, it must be that $S_{v_i \leftrightarrow w_j}, S_{v_i \leftrightarrow w_\ell} \in T_V$ due to the covering lemma and the fact that the base gadget is compatible ISS pair. This contradicts our previous argument about V-sets. □

Now, we finish the proof as follows. Let $V_T \subseteq V$ be the vertices that have their V-sets in T. Note that V_T is well defined due to the above lemma, and further, $|V_T| = k$. We claim that V_T is isomorphic to k-CLIQUE H. Consider $v_i \neq v_{i'} \in V_T$. Let $S_{(v_i \leftrightarrow w_j)}$ and $S_{(v_{i'} \leftrightarrow w_{j'})}$ be the corresponding V-sets in T_V, for some $w_j \neq w_{j'} \in W$. Then, note that, using Claim 6.5.1, it must be that $S_{(v_i,v_{i'}) \leftrightarrow (w_j,w_{j'})} \in T_E$. Hence, we have $(v_i, v_{i'}) \in E$. □

6.6 Conclusion and Open Problems

The research presented in Publication IV lays the foundation for the parameterized complexity of combinatorial problems in a compact regime, exploring aspects related to COMPACT PARAMETERIZED SET PACKING (COMPACT PSP) and COMPACT r-VECTORSUM. The work establishes a dichotomy theorem for the parameterized complexity of COMPACT PSP by identifying a combinatorial gadget called Intersecting Set System

Pair (ISS) that is crucial in obtaining **W[1]**-hardness for COMPACT PSP. An interesting direction is FPT approximating COMPACT PSP: Given a promise that there is an r-packing, is it possible to find a packing of size $\omega(1)$ in FPT time? Note that for the general PSP problem, there is no $o(r)$ FPT-approximation, assuming Gap-ETH. However, recent results [100, 114] use a weaker assumption of $\mathbf{W[1]} \neq \mathsf{FPT}$ but also obtain weaker FPT-inapproximibility. It is also interesting to show such a hardness of approximation for COMPACT PSP. The research also improves the lower bound for COMPACT r-VECTORSUM. An interesting avenue for further investigation is to explore whether this lower bound can be further tightened.

7. k-claw-free Graphs: Conditional χ-boundedness and Convex Relaxations for Independent Set

The chapter is based on Publication V [37] and focuses on the computational complexity of the structural properties of connected k-claw-free graphs, exploring the connection between an extremal combinatorics question and the power of a convex program in approximating the maximum-weight independent set in this graph class. For the extremal question, we consider the notion, that we call *conditional χ-boundedness* of a graph: Given a graph G that is assumed to contain an independent set of a certain (constant) size, we are interested in upper bounding the chromatic number in terms of the clique number of G. This question, besides being interesting on its own, has algorithmic implications (which have been relatively neglected in the literature) on the performance of semi-definite program relaxations in estimating the value of maximum-weight independent set.

For $k = 3$, Chudnovsky and Seymour [49] prove that any 3-claw-free graph G with an independent set of size three must satisfy $\chi(G) \le 2\omega(G)$. Their result implies a factor 2-estimation algorithm for the maximum weight independent set via an SDP relaxation (providing the first non-trivial result for maximum-weight independent set in such graphs via a convex relaxation). An obvious open question is whether a similar conditional χ-boundedness phenomenon holds for any k-claw-free graph. In this chapter, we explore this question in depth. Additionally, we establish connections between the previously mentioned extremal question and the power of convex relaxations.

7.1 χ-boundedness and Convex Relaxations

This chapter focuses on the following problem.

Definition 7.1.1 (Maximum Weight Independent Set (MWIS)). *Given a graph $G = (V, E)$ and a weight function $w : V \to \mathbb{R}_{\geq 0}$, the maximum weight independent set problem asks to find an independent set in G of maximum weight.*

Recall that $\alpha(G)$ denotes the maximum weight of an independent set in G. Therefore, the MWIS problem asks to find an independent set whose weight is $\alpha(G)$. Now, there is a related, simpler problem called the *Estimating Maximum Weight Independent Set*. In this problem, the objective is to compute $\alpha(G)$ itself. In this chapter, we will delve into the challenges of computing $\alpha(G)$ and solving the MWIS problem when the graph G is a connected k-claw-free graph. It is worth noting that for this case, a basic greedy algorithm that selects an arbitrary maximal independent set is capable of achieving an approximation factor of $(k-1)$.

QSTAB. Given an unweighted graph $G=(V,E)$ with $V=\{1,2,\cdots,n\}$, consider the following polytope defined by the linear inequalities.

$$\mathsf{QSTAB}(G)=\{x\in[0,1]^n \mid \sum_{i\in Q} x_i \le 1 \quad \forall \text{clique } Q\} \tag{7.1}$$

It is also known as clique constraint stable set polytope. Note that maximizing the function $f(x)=\sum_{i\in V} x_i$ over $\mathsf{QSTAB}(G)$ yields an upper bound on $\alpha(G)$. Further, this convex relaxation can be optimized in polynomial time via semi-definite programs.

QSTAB has been shown to perform relatively well in terms of approximating many crucial problems, such as set packing [59, 43, 82, 83] and independent set in sparse graphs [18, 11]. Despite these successful cases, QSTAB fails unexpectedly in graph classes such as k-claw-free graphs: For any $k\ge 3$, a simple greedy algorithm immediately gives a factor $(k-1)$ approximation for MWIS, while QSTAB is unable to give $f(k)$ estimation for any function f (see the full version of our paper [37]).

A closely related concept to MWIS in graph theory is known as χ-bounded graphs. A graph G is said to be *χ-bounded* if $\chi(G)\le f(\omega(G))$, for some function f. This concept has been extensively studied in graph theory (see the comprehensive survey [117] and references therein). When the function f is linear, we say that G is linearly χ-bounded.

The following theorem of [38] establishes a connection between the computational power of QSTAB to the extremal question of χ-boundedness.

Theorem 7.1.1 ([38]). *Let $\mathcal{G}$ be a graph class closed under clique replacement*[1]*. Then, for any $G\in\mathcal{G}$, the integrality gap of* $\mathsf{QSTAB}(G)$ *is at most γ if and only if $\chi(G)\le\gamma\omega(G)$.*

This theorem, in combination with the observation that QSTAB exhibits arbitrarily large integrality gap on k-claw-free graphs [37], provides a definitive answer to the question of whether k-claw-free graphs are linearly χ-bounded. The answer is negative, meaning that k-claw-free graphs do not belong to the linearly χ-bounded graph class.

[1]A clique replacement operation on graph G replaces any vertex v with a clique K_v of arbitrary size and connects each vertex in a clique to every neighbor of v. It is easy to see that k-claw-free graphs are closed under clique replacement.

7.2 Towards Stronger Relaxations

A further way to estimate $\alpha(G)$ in k-claw-free graphs is to investigate the integrality gap of more stronger relaxations than QSTAB. In this chapter, we explore two such relaxations.

(i) Sherali–Adams strengthening of QSTAB

(ii) Sum-of-Squares (SoS) hierarchy, a more stronger relaxation than (i)

To better comprehend our results, we will introduce some necessary terminology. However, for a complete exposition on sum-of-squares and related convex relaxations, we refer the readers to excellent survey papers [95, 65, 113].

For a graph $G=(V,E)$, let K_G be the polytope $\{x \in [0,1]^V : x_i + x_j \leq 1, \forall (i,j) \in E\}$. The standard LP relaxation for MWIS $\max\{\sum_v w(v)x_v : x \in K_G\}$ is known to have integrality gap of at least $\Omega(n)$ on n-vertex graphs[2].

7.2.1 Sherali–Adams Strengthening of QSTAB

Sherali–Adams hierarchies [118] is a family of increasingly tight convex relaxations. Let $G=(V,E)$ be a graph with $V=[n]$ of (unweighted) maximum independent set problem. Recall the QSTAB LP for G.

$$\begin{aligned}
(\mathsf{QSTAB}\ (G)) \quad \max \quad & \sum_{i\in[n]} x_i \\
\text{s.t.} \quad & 1-\sum_{i\in Q} x_i \geq 0 \quad \forall \text{ clique } Q \\
& x_i \geq 0 \quad \forall i \in [n]
\end{aligned}$$

For $\ell \geq 1$, the Sherali Adams hierarchy applied on QSTAB(G) is as follows.

$$(\mathsf{SA}^+_\ell(G)) \qquad \max \sum_{i\in[n]} y_{\{i\}}$$

$$\text{s.t.} \forall S,T \subseteq [n], S\cap T=\emptyset, |S\cup T| \leq \ell \text{ following holds.}$$

$$\sum_{T'\subseteq T}(-1)^{|T'|} y_{S\cup T'} - \sum_{i\in Q}\sum_{T'\subseteq T}(-1)^{|T'|} y_{S\cup T'\cup\{i\}} \geq 0 \ \forall \text{ clique } Q \tag{7.2}$$

$$\sum_{T'\subseteq T}(-1)^{|T'|} y_{S\cup T'} \geq 0 \tag{7.3}$$

$$\sum_{T'\subseteq T}(-1)^{|T'|} y_{S\cup T'\cup\{i\}} \geq 0 \quad \forall i \in [n] \tag{7.4}$$

$$y_\emptyset = 1 \tag{7.5}$$

[2]Consider the clique K_n on n vertices and LP assignment $x_i = 1/2$ for vertex $i \in K_n$.

7.2.2 Sum-of-Squares

The sum-of-square hierarchies (or Lasserre hierarchies) [93, 94, 109, 110] can be applied to (increasingly) tighten any linear program (captured by the level in the hierarchy). For any $t \geq 1$, the t-th level of SoS can be computed in time $n^{O(t)}$.

Now we formally define SoS, following the treatment of Rothvoss [113].

Definition 7.2.1. *Define the t-th level of SoS hierarchy* $\mathsf{SoS}_t(K_G)$ *as the set of vectors* $z \in \mathbb{R}^{2^{V(G)}}$ *that satisfy:*

$$M_t(z) := (z_{I \cup J})_{|I|,|J| \leq t} \succeq 0 \qquad \text{and}$$

$$M_t^{ij}(z) := (z_{I \cup J} - z_{I \cup J \cup \{i\}} - z_{I \cup J \cup \{j\}})_{|I|,|J| \leq t} \succeq 0 \quad (\forall (i,j) \in E(G))$$

Let $\mathsf{SoS}_t^{proj}(K_G) = \{(z_i)_{i \in V(G)} : z \in \mathsf{SoS}_t(K_G)\}$ *be the projection on the original variables.*

It is standard to view 1 as $z_\emptyset = 1$ (so that we have variables z_I for all subsets).

Proposition 7.2.1 (Lemma 8 in [113])**.** *If any solution* $x \in K_G$ *contains at most t ones, then any* $z \in \mathsf{SoS}_t^{proj}(K_G)$ *is a convex combination of integer solutions in* $K_G \cap \{0,1\}^{V(G)}$.

Corollary 7.2.1. *Let* $z^* = \arg\max\{\sum_{i \in V(G)} w(i) z_i : z \in \mathsf{SoS}_t(K_G)\}$. *If* $\alpha(G) \leq t$, *then the objective value of* z^* *is exactly the value of maximum weight independent set in G.*

Our next proposition states that any feasible solution of SoS (at level at least two) can be projected into a feasible solution for QSTAB. The proof is somewhat of a folklore nature. Since it has never been written anywhere in the form we need, we provide a proof in our full version [37] for completeness.

Proposition 7.2.2. *Let* $z \in \mathsf{SoS}_t^{proj}(G)$ *for* $t \geq 2$. *Then* $z \in \mathsf{QSTAB}(G)$.

It can be shown that $\mathsf{SoS}_{\ell+2}$ is at least as strong as $\mathsf{SA}_\ell^+(G)$. The formal statement is encapsulated in the following proposition.

Proposition 7.2.3. *Let* $z \in \mathsf{SoS}_{\ell+2}(K_G)$. *Then the solution* $\{z_I\}_{|I| \leq \ell}$ *is feasible for* $\mathsf{SA}_\ell^+(G)$.

7.2.3 Conditional χ-boundedness and SoS

Beyond its superior strength compared to Sherali-Adams, the Sum-of-Squares (SoS) relaxation also addresses a more refined extremal question, specifically, about conditional χ-boundedness, which we explain in this section.

Conditional χ-boundedness We say that G is (t,γ)-conditionally χ-bounded if $\chi(G) \le \gamma\omega(G)$ whenever $\alpha(G) \ge t$. Moreover, a graph class is (t,γ)-conditionally χ-bounded if every graph in that class is.

In their seminal work on claw-free graphs, Chudnovsky and Seymour [49] showed that $\chi(G) \le 2\omega(G)$ when G is a claw-free graph with $\alpha \ge 3$. For general k, such bounds, when f is a function of k, have fruitful algorithmic implications on MWIS. The following theorem (which is a simple combination of known facts) from Publication V connects the power of SoS to an extremal question of Chudnovsky and Seymour.

Theorem 7.2.1. *Consider a graph class that is closed under clique replacement. If the graph class is (t,γ)-conditionally χ-bounded, then t rounds of SoS gives a factor γ-estimation for MWIS.*

Thus, using Theorem 7.2.1, the aforementioned result of Chudnovsky and Seymour [49] gives an efficient 2-estimation algorithm for MWIS in claw-free graphs.

Especially, if $t = O(1)$, then SoS gives a γ-estimation algorithm for MWIS in polynomial time. Therefore, it raises an intriguing algorithmic question: Can SoS provide an approximate estimate of $\alpha(G)$ in k-claw-free graph G in constant rounds, even if the approximation factor is $f(k)$, where f is some function?

7.3 Related Work

The MWIS problem on k-claw-free graphs is notoriously hard, even in the unweighted case. As for approximating this problem, a local search algorithm due to Berman [22] has remained the best-known approximation algorithm for the past two decades. Berman's algorithm achieved a factor of $\frac{k}{2} + \epsilon$ in polynomial time. In 2021, Neuwohner [106] broke the barrier of $\frac{k}{2}$ by improving the factor to $\frac{k}{2} - \frac{1}{63{,}700{,}992}$. In a very recent work, Thiery and Ward [124] improved the factor to $\frac{k}{2} - \delta_k$ for a constant $\delta_k \ge 0.214$. On the other hand, the problem is NP-hard to approximate better than a factor of $\Omega(k/\log k)$ [82][3]. With respect to convex programs, Chan and Lau [41] study the power of standard LP and Sherali–Adams on k-SET PACKING, which is a special case of maximum independent set in $(k+1)$-claw-free graph. In k-SET PACKING, we are given a k-uniform hypergraph H on n vertices, and we are asked to find a maximum matching in H. In their work, [41] show that Sherali–Adams on the standard LP for k-SET PACKING has an integrality gap of at least $k-2$, even after $\Omega(n/k^3)$ rounds. Additionally, they show that, for constant k, QSTAB for k-SET PACKING can be captured by a polynomial size LP and has an integrality gap of at most $(k+1)/2$.

[3] In fact, the hardness even holds for a special case of the problem, namely the unweighted k-set packing problem.

7.4 Thesis Contribution

Our first result, Theorem 7.2.1 mentioned above, bounds the integrality gap of SoS for maximum independent set on graph classes with a new notion of conditional χ-boundedness. In light of Theorem 7.2.1, let us state the theorem of Chudnovsky and Seymour for claw-free graphs.

Theorem 7.4.1 (Chudnovsky-Seymour (CS theorem) [49]). *For every connected claw-free ($k = 3$) graph G with $\alpha(G) \geq 3$, it holds that $\chi(G) \leq 2\omega(G)$.*

Hence, combining the above theorem with our Theorem 7.2.1, we obtain that SoS with only 3 rounds provides factor 2-estimation on the maximum weighted independent set on claw-free graphs. Our first main result is to rule out this approach for general $k \geq 4$. In particular, we show the following graph theoretic result that refutes the possibility of generalizing the above CS theorem.

Theorem 7.4.2. *For $k \geq 4$, there exists n_0 depending on k such that for infinitely many $n \geq n_0$, there exists a connected k-claw-free graph G on n vertices with $\alpha(G) = \Omega\left(\frac{n}{\log n}\right)$ and $\chi(G) \geq f(k)\left(\frac{\omega(G)}{\log \omega(G)}\right)^{k/2}$, for some $f(k)$.*

We remark that our lower bound result almost matches the upper bound of [129], and both the upper and lower bounds are tight with respect to the state-of-the-art Ramsey bounds (in the sense that improving them requires improving Ramsey bounds).

Consequently, an obvious question arises: Is there a lower bound on the integrality gap of the SoS relaxation that demonstrates its inferior performance compared to a straightforward greedy algorithm, which achieves a $(k-1)$-factor approximation for this problem? Our second main result makes progress in this direction. Particularly, in the following result, we construct a bad example using our counter example of Theorem 7.4.2 for *Sherali-Adams* strengthening of QSTAB. In the following, we use the notations O_k, Θ_k, Ω_k to hide multiplicative functions in k.

Theorem 7.4.3 (Integrality gap of Sherali-Adams on QSTAB). *Let $0 < \epsilon \leq 1/3$. For $k \geq 4$, for infinitely many n, there exists a connected k-claw-free graph G_n on n vertices with $\alpha(G_n) = \Theta_k(n^{\epsilon})$ such that the integrality gap of Sherali-Adams hierarchy on QSTAB(G_n) is $\Omega_k(n^{\epsilon})$, even after $\Omega_k(n^{1-2\epsilon})$ rounds.*

To contrast the above theorem with the performance of Sherali-Adams on special cases of k-claw-free graphs, such as the maximum independent set in degree d graphs, note that in degree d graphs, Sherali-Adams is known to provide a tight approximation factor [18] within only poly-logarithmic rounds in d (assuming Unique Games Conjecture (UGC) [11]).

Discussion of previous work on convex programs. Let us compare our result of Theorem 7.4.3 with the bounds of Chan and Lau [41]. In particular, [41] considers k-Set Packing, a special case of maximum independent set in $(k+1)$-claw-free graph, where we are given a k-uniform hypergraph H on n vertices and we are asked to find a maximum matching in H. In their work, [41] show that Sherali–Adams on the standard LP for k-SET PACKING has an integrality gap of at least $k-2$, even after $\Omega_k(n)$ rounds. Additionally, they show that, for constant k, QSTAB for k-SET PACKING can be captured by a polynomial size LP and has an integrality gap of at most $(k+1)/2$. In contrast, our result of Theorem 7.4.3 is for a more general problem of maximum independent set in k-claw-free graphs and yields an integrality gap of $\Omega_k(n^\epsilon)$, which is a function of n, for Sherali–Adams on QSTAB with rounds $\Omega_k(n^{1-2\epsilon})$, which is a stronger program than that of [41]. However, we remark that the results of Theorem 7.4.3 can be adjusted to yield an integrality gap of $g(k)$ for any function g for Sherali–Adams on QSTAB with $\Omega_k(n)$ number of rounds, which is linear in n. Thus, our results yield a larger integrality gap for a stronger program for maximum independent set in k-claw-free graphs compared to that of [41].

7.5 Preliminaries

We follow standard graph theoretic notation. Given a graph $G = (V(G), E(G))$, $M \subseteq E(G)$ is a *matching* if no pair of edges in M share a vertex. Further, a matching $M \subseteq E(G)$ is said to be an *induced matching* if it is an induced subgraph of G. Finally, for a matching M, $e_i \neq e_j \in M$ is said to be an *intersecting matching pair* if e_i and e_j do not form an induced matching, and M is an *intersecting matching* if every pair in M is an intersecting matching pair.

For $k \geq 3$, a *k-claw* is the graph $K_{1,k}$. For a k-claw T, the vertex with degree k is called the *central vertex* of T, and the remaining vertices of degree one are called *leaves* of T. A graph G is said to be *k-claw-free* if there exists no k-claw as an induced subgraph. For a graph G, $\alpha(G)$ is the size of maximum independent set in G, $\omega(G)$ is the size of maximum clique in G, and $\chi(G)$ is the chromatic number of G. For weighted case, $\alpha(G)$ and $\omega(G)$ represent the maximum weight of an independent set and the maximum weight of a clique in G, respectively.

The notations O_k, Θ_k, Ω_k hide multiplicative functions in k.

7.6 Overview of Techniques

In this section, we will show the key ideas and techniques involved in proving Theorem 7.4.2. Our graph for the proof is based on Ramsey graphs and a novel graph composition technique.

Ramsey Graphs. Let $R(s,t)$, for $s \geq 3$, denote the Ramsey number, i.e., $R(s,t)$ is the minimum number such that any graph on $R(s,t)$ vertices has either an independent set of size s or a clique of size t.

Theorem 7.6.1 ([87, 27, 28]). *For any $s \geq 3$ and for sufficiently large t, $R(s,t) \geq c'_s \cdot t^{\frac{s+1}{2}} (\log t)^{\frac{1}{s-2} - \frac{s+1}{2}}$, for some constant c'_s depending only on s.*

Thus, the above theorem implies that for $s \geq 3$ and sufficiently large t, there is a graph on $\lceil R(s,t) - 1 \rceil$ vertices that has neither an independent set of size s nor a clique of size t. We call such a graph an *(s,t)-Ramsey graph*.

Corollary 7.6.1. *For $s \geq 3$ and sufficiently large t, there is an (s,t)-Ramsey graph on $c_s \cdot t^{\frac{s+1}{2}} (\log t)^{\frac{1}{s-2} - \frac{s+1}{2}}$ vertices for some positive constant c_s depending only on s.*

The key ingredient of our construction is based on our new graph composition, which we call *bi-conflict composition*.

Bi-conflict Composition. Given a graph H, the bi-conflict composition of H with itself is a graph $B := H \circledb H$ obtained as follows. First, create an auxiliary graph D from two copies H^1 and H^2 of H, and add an edge between every vertex in H^1 and its corresponding vertex in H^2. We call such edges *matching edges* of D. Then, we create a vertex b_e in B for every matching edge e of D and add an edge $(b_e, b_{e'}) \in E(B)$ if e and e' form an interesting matching pair in D.

Graph Construction. We first show a simple attempt that, while falling short of achieving the result stated in Theorem 7.4.2, offers valuable insights and promising ideas for potential improvements. Our starting point is Theorem 7.6.1, which yields a graph H on n vertices such that $\alpha(H) = 2$ and $\omega(H) \leq c\sqrt{n \log n}$. Note that H is 3-claw-free and $\chi(H) \geq n/\alpha(H) = n/2 \geq \tilde{\Omega}(\omega(H)^2)$. Hence, H has all the desired properties of Theorem 7.4.2 for $k = 3$, except that $\alpha(H)$ is very small. To boost the independent set, we will use bi-conflict composition on H four times to get graph D. Let H^1, H^2, H^3, and H^4 be copies of H. Then, the graph D is the union of graphs $H^1 \circledb H^2, H^2 \circledb H^3$ and $H^3 \circledb H^4$. Note that the following holds true for D.

- D is 5-claw-free,
- $\alpha(D) = 4$, and
- $\omega(D) \leq 3c\sqrt{n \log n}$.

If we extend the same idea and compose H ℓ times, we can maintain the properties that D remains 5-claw-free with $\ell \leq \alpha(D) \leq 2\ell$ and $\omega(D) \leq 3c\sqrt{n \log n}$, thus achieving the desired results for $k = 4$. However, to generalize this construction for any $k \geq 4$ and to obtain tighter bounds as stated in Theorem 7.4.2, we will need to explore further ideas, which we elaborate on next.

Lemma 7.6.1. *Let $k \geq 4, p \geq 1, \tau \geq 3$, and let $H = (V(H), E(H))$ be a $(k-1, \tau)$-Ramsey graph. Then, there exists a connected k-claw-free graph G on $\Theta(p|V(H)|)$ vertices such that $p \leq \alpha(G) \leq 3pk$ and $\omega(G) \leq 3\tau$.*

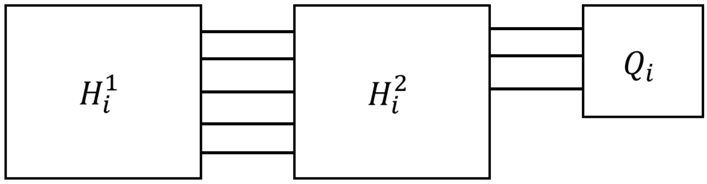

Figure 7.1. One block B_i of the graph G'

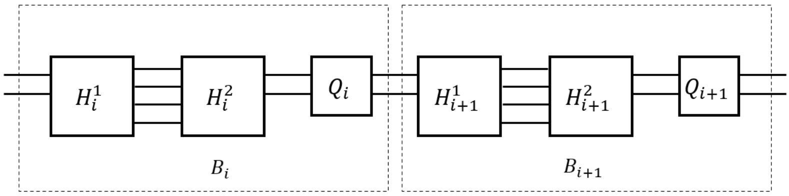

Figure 7.2. Connecting block B_i and B_{i+1} in G'

Proof. We construct the graph G in two steps. In the first step, we construct an auxiliary graph $G' = (V(G'), E(G'))$ using $(k-1,\tau)$-Ramsey graph H. Finally, in the second step, using this auxiliary graph G', we construct our graph G.

Construction of auxiliary graph G'. Let $Q = K_{\tau-1}$ be the complete graph on $\tau-1$ vertices. We construct the graph G' in two steps: In the first step, we use H and Q to create a graph B, which we call a *block*. In the second step, we construct the graph G' using p copies of B. For $i \in [p]$, we construct block B_i using Q and two copies of H as follows (see Figure 7.1). Let H_i^1 and H_i^2 be copies of H, and let Q_i be a copy of Q. We connect each vertex in H_i^1 with its respective copy in H_i^2 by an edge. Then, we pick an arbitrary set of $\tau-1$ vertices from H_i^2 and add an (arbitrary) matching between this set and the vertices of Q_i. Finally, we connect each B_i to $B_{(i+1) \bmod p}$ as follows (see Figure 7.2). We pick an arbitrary set of $\tau-1$ vertices from $H^1_{(i+1) \bmod p}$ and add a matching between this set and the vertices of Q_i. This completes the construction of G'. Notice that G' forms a ring structure consisting of B_is. We call the new edges that we added in our construction *matching edges*.

Construction of G. For every matching-edge e_i in G', we create a vertex v_i in G. Then, $(v_i, v_j) \in E(G)$ if e_i and e_j is an intersecting matching pair (i.e., they do not form an induced matching).

Analysis. Let $n = |V(G)|$, then note that $n = p(q_k(\tau) + 2(\tau-1))$. Hence, $3pq_k(\tau) > n \geq pq_k(\tau)$. Since the Ramsey graph H and $Q = K_{\tau-1}$ used in the construction of B are connected, we have that G' is connected. This implies that G is connected. Next, we bound $\alpha(G)$. First note that $\alpha(G) \geq p\alpha(H)$

since from each block B_i we can pick matching edges between H_i^1 and H_i^2 that have endpoints on the vertices of $\alpha(H)$. The vertices in G corresponding to these matching edges form an independent set. For the upper bound, we will show that $\alpha(G) < 3p\alpha(H)$. Suppose, for contradiction, $\alpha(G) \geq 3p\alpha(H)$. Let I be an independent set in G with $|I| \geq 3p\alpha(H)$. Let M be the set of matching-edges of G' corresponding to the vertices in I. Let $M_i \subseteq M$ be the edges of M with both endpoints in B_i. Since B_i and $B_{(i+1) \bmod p}$ are connected by Q_i (which is a complete graph), the number of edges of M that are between B_is is at most p. Hence, the number of edges of M that lie completely within some B_i is at least $M-p$. Since $|M| \geq 3p\alpha(H)$, it must be that there is a block B_i such that $|M_i| \geq (3p\alpha(H)-p)/p \geq 2\alpha(H)$. Since the edges of M_i should have an endpoint in H_i^2, it implies that $\alpha(H_i^2) \geq |M_i| \geq 2\alpha(H)$, which is a contradiction since H_i^2 is a copy of H. Thus, we have that $p\alpha(H) \leq \alpha(G) < 3p\alpha(H)$.

For bounding $\omega(G)$, let $\mathscr{C}$ be a clique of maximum size in G. We claim that $|\mathscr{C}| \leq 3\tau$, which implies that $\omega(G) \leq 3\tau$. Let $E_{\mathscr{C}}$ be the matching-edges of G' corresponding to $\mathscr{C}$. Let $E'_{\mathscr{C}} \subseteq E_{\mathscr{C}}$ be the edges that have one endpoint in some copy of Q, and let $E''_{\mathscr{C}} = E_{\mathscr{C}} \setminus E'_{\mathscr{C}}$. Suppose $E'_{\mathscr{C}} \neq \emptyset$, then consider an edge $e' \in E'_{\mathscr{C}}$ and suppose e' is incident on Q_i some for $i \in [p]$. Then, observe that every edge in $E'_{\mathscr{C}}$ must also be incident on Q_i since $E'_{\mathscr{C}}$ is an intersecting matching and edges incident on Q_i cannot intersect with edges incident on $Q_j, j \neq i \in [p]$ by construction. Thus, $|E'_{\mathscr{C}}| \leq 2|Q| < 2\tau$. Hence, when $E''_{\mathscr{C}} = \emptyset$, we have that $|E_{\mathscr{C}}| < 2\tau$, implying $|\mathscr{C}| < 2\tau$, as desired. For the other case, when $E''_{\mathscr{C}} \neq \emptyset$, then consider $e'' \in E''_{\mathscr{C}}$. We will bound the number of neighbors of e'' in $E'_{\mathscr{C}}$ and $E''_{\mathscr{C}}$. First, note that e'' has at most $|Q| < \tau$ neighbors in $E'_{\mathscr{C}}$ since e'' can intersect with at most $|Q| = \tau - 1$ many edges from Q_i. Next, we will show that e'' has at most $2(\tau-1)$ neighbors in $E''_{\mathscr{C}}$, which implies that $|E''_{\mathscr{C}}| < 2\tau$, and hence $|\mathscr{C}| = |E_{\mathscr{C}}| = |E'_{\mathscr{C}}| + |E''_{\mathscr{C}}| < 3\tau$, as desired. To see this, note that e'' is incident on two adjacent copies of H, and hence the number of (matching) edges between these copies of H that form an intersecting matching together with e'' is at $2\omega(H) = 2(\tau-1)$.

Finally, we show that G is k-claw-free. Suppose, for contradiction, there is a k-claw T in G with central vertex $v \in T$ and leaves $v_1, \cdots, v_k \in T$. Let $M_T = \{e, e_1, \cdots, e_k\}$ be the matching-edges in G' corresponding to T with e corresponding to v and e_i corresponding to v_i, for $i \in [k]$. Also, let $L_T = \{e_1, \cdots, e_k\}$. First consider the case when e is incident between a copy H' of H and a copy Q' of Q. Let $I \subseteq V(H')$ be the endpoints of edges of L_T in H'. We claim that $|I| \geq k-1$. To see this, note that for every edge $e_j \in L_T$ that is not incident on Q' must have one endpoint in H' since e_j and e form an intersecting matching pair. On the other hand, since Q' is a clique, there can be at most one edge in L_T with an endpoint in Q', implying $|I| \geq k-1$. As edges in L_T form an induced matching in G', we have that I is an independent set in H'. But then $|I| \geq k-1$, which is a contradiction since $\alpha(H') < k-1$. Now, consider the other case when e is between two copies of H' and H'' of H. Let $L'_T \subseteq L_T$ be the edges that are between H' and H'', and let $L''_T = L_T \setminus L'_T$. Then, note that $|L'_T| < |L_T| = k$, since otherwise the endpoints of L'_T in H' (or H'') form an independent set of size k in H' (or H'' resp.), leading to the contradiction to the fact that $\alpha(H) < k-1$. Suppose $(k-2) \leq |L'_T| \leq (k-1)$, and let $e_j \in L''_T$. Then, since e_j and e form an intersecting matching pair, assume, without loss of generality, that e_j has an endpoint in H'. But then, since every edge of $L'_T \cup \{e_j\}$ has one endpoint in H', the endpoints of $L'_T \cup \{e_j\}$ in H' form an independent set of size $k-1$ in H' , leading to a contradiction. Finally, if $|L'_T| < k-2$, then there must be at least two edges $e_i, e_j \in L''_T$ incident on one of the two copies of Q adjacent to H' and H''. But since Q is a clique, this means $(v_i, v_j) \in E(G)$, contradicting the fact that T is a k-claw. □

Proof of Theorem 7.4.2. For $k \geq 4$, let H be a $(k-1,t)$-Ramsey graph obtained from Corollary 7.6.1, for every sufficiently large t. Let $q_k(t) = |V(H)| = c_k t^{\frac{k}{2}} (\log t)^{\frac{1}{k-3}-\frac{k}{2}}$. Then, using $p = 2^{q_k(t)}$ and $\tau = t$ along with graph H, Lemma 7.6.1 produces a graph G on $n := \Theta(2^{q_k(t)} q_k(t))$ vertices such that $\alpha \geq 2^{q_k(t)} = \Omega(n/\log n)$, and $\alpha(G) \leq 3k 2^{q_k(t)} = O(nk/\log n)$, and $\omega(G) \leq 3t$. Hence, we have

$$\chi(G) \geq \frac{n}{\alpha(G)} = \Omega\left(\frac{q_k(t)}{k}\right) = \Omega\left(\frac{c_k}{3k}\left(\frac{t}{\log t}\right)^{k/2}\right) = f(k)\left(\frac{\omega(G)}{\log \omega(G)}\right)^{k/2},$$

for some $f(k)$. □

7.7 Conclusion and Open Problems

The work presented in Publication V marks the initiation of research into estimating the size of the maximum weight independent set in k-claw-free graphs using convex relaxations. A central question in this line of inquiry revolves around identifying whether there exists a convex relaxation that can offer a reasonable estimate for this problem. To tackle this challenge, the paper establishes a conceptual link between the Chudnovsky-Seymour theorem of conditional χ-boundedness and the integrality gap of Sum-of-Squares (SoS) relaxations. Moreover, for the general case of $k \geq 4$, the paper introduces a family of k-claw-free graphs that pose challenges when attempting to generalize such conditional χ-boundedness results. Additionally, there is evidence to suggest that this particular graph family may present challenges for the Sum-of-Squares (SOS) hierarchy.

However, an important question remains unanswered by this research: Can the Sum-of-Squares (SoS) approach match the approximation guarantees offered by the simple greedy algorithm for this problem? This question remains an open and intriguing area of investigation for future research.

References

[1] Fairclustering.com.

[2] Fateme Abbasi, Sandip Banerjee, Jarosław Byrka, Parinya Chalermsook, Ameet Gadekar, Kamyar Khodamoradi, Dániel Marx, Roohani Sharma, and Joachim Spoerhase. Parameterized approximation for robust clustering in discrete geometric spaces, 2023.

[3] Fateme Abbasi, Sandip Banerjee, Jarosław Byrka, Parinya Chalermsook, Ameet Gadekar, Kamyar Khodamoradi, Dániel Marx, Roohani Sharma, and Joachim Spoerhase. Parameterized approximation schemes for clustering with general norm objectives, 2023.

[4] Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. Fair clustering via equitable group representations. In Madeleine Clare Elish, William Isaac, and Richard S. Zemel, editors, *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, pages 504–514. ACM, 2021.

[5] Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. Fair clustering via equitable group representations. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT'21)*, pages 504–514, 2021.

[6] Marek Adamczyk, Jarosław Byrka, Jan Marcinkowski, Syed M Meesum, and Michał Włodarczyk. Constant-factor FPT approximation for capacitated k-median. In *27th Annual European Symposium on Algorithms (ESA'19)*, page 1. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, 2019.

[7] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Mach. Learn.*, 75(2):245–248, may 2009.

[8] Barbara Anthony, Vineet Goyal, Anupam Gupta, and Viswanath Nagarajan. A plant location guide for the unsure: Approximation algorithms for min-max location problems. *Mathematics of Operations Research*, 35(1):pages 79–101, 2010.

[9] Giorgio Ausiello, Alessandro D'Atri, and Marco Protasi. Structure preserving reductions among convex optimization problems. *Journal of Computer and System Sciences*, 21(1):136–153, 1980.

[10] Per Austrin, Subhash Khot, and Muli Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 74–80. IEEE, 2009.

[11] Per Austrin, Subhash Khot, and Muli Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. *Theory of Computing*, 7(1):27–43, 2011.

[12] Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The hardness of approximation of euclidean k-means. In *31st International Symposium on Computational Geometry (SoCG'15)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[13] Mihai Bâdoiu and Kenneth L. Clarkson. Smaller core-sets for balls. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, page 801–802, USA, 2003. Society for Industrial and Applied Mathematics.

[14] Mihai Badŏiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In John H. Reif, editor, *Proc. 34th Annual ACM Symposium on Theory of Computing (STOC'02)*, pages 250–257. ACM, 2002.

[15] Tanvi Bajpai, Deeparnab Chakrabarty, Chandra Chekuri, and Maryam Negahbani. Revisiting priority k-center: Fairness and outliers. In *48th International Colloquium on Automata, Languages, and Programming (ICALP'21)*, page 21. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[16] Daniel Baker, Vladimir Braverman, Lingxiao Huang, Shaofeng H-C Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for clustering in graphs of bounded treewidth. In *International Conference on Machine Learning (ICML'20)*, pages 569–579. PMLR, 2020.

[17] Daniel Baker, Vladimir Braverman, Lingxiao Huang, Shaofeng H.-C. Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for clustering in graphs of bounded treewidth. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 569–579. PMLR, 13–18 Jul 2020.

[18] Nikhil Bansal, Anupam Gupta, and Guru Guruganesh. On the Lovász theta function for independent sets in sparse graphs. In *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*, pages 193–200, 2015.

[19] Nikhil Bansal, Anupam Gupta, and Guru Guruganesh. On the lovász theta function for independent sets in sparse graphs. *SIAM Journal on Computing*, 47(3):1039–1055, 2018.

[20] Nikhil Bansal, Rohit Khandekar, Jochen Könemann, Viswanath Nagarajan, and Britta Peis. On generalizations of network design problems with degree bounds. *Mathematical Programming*, 141(1):479–506, 2013.

[21] Suman Kalyan Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS'19)*, pages 4955–4966, 2019.

[22] Piotr Berman. A $d/2$ approximation for maximum weight independent set in d-claw free graphs. In Magnús M. Halldórsson, editor, *Algorithm Theory - SWAT 2000, 7th Scandinavian Workshop on Algorithm Theory, Bergen, Norway, July 5-7, 2000, Proceedings*, volume 1851 of *Lecture Notes in Computer Science*, pages 214–219. Springer, 2000.

[23] Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Faster algorithms for the constrained k-means problem. *Theory of Computing Systems*, 62(1):93–115, 2018.

[24] Sayan Bhattacharya, Parinya Chalermsook, Kurt Mehlhorn, and Adrian Neumann. New approximability results for the robust k-median problem. In *Scandinavian Workshop on Algorithm Theory (SWAT'14)*, pages 50–61. Springer, 2014.

[25] Arnab Bhattacharyya, Ameet Gadekar, Suprovat Ghoshal, and Rishi Saket. On the Hardness of Learning Sparse Parities. In Piotr Sankowski and Christos Zaroliagis, editors, *24th Annual European Symposium on Algorithms (ESA 2016)*, volume 57 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:17, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[26] Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set partitioning via inclusion-exclusion. *SIAM J. Comput.*, 39(2):546–563, July 2009.

[27] Tom Bohman. The triangle-free process. *Advances in Mathematics*, 221(5):1653–1677, 2009.

[28] Tom Bohman and Peter Keevash. The early evolution of the h-free process. *Inventiones mathematicae*, 181(2):291–336, 2010.

[29] Vladimir Braverman, Shaofeng H-C Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for ordered weighted clustering. In *International Conference on Machine Learning (ICML'19)*, pages 744–753. PMLR, 2019.

[30] Vladimir Braverman, Shaofeng H-C Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for clustering in excluded-minor graphs and beyond. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA'21)*, pages 2679–2696. SIAM, 2021.

[31] Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Trans. Algorithms*, 13(2), mar 2017.

[32] Jarosław Byrka, Krzysztof Sornat, and Joachim Spoerhase. Constant-factor approximation for ordered k median. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC'18)*, pages 620–631, 2018.

[33] Liming Cai and David Juedes. Subexponential parameterized algorithms collapse the w-hierarchy. In *International Colloquium on Automata, Languages, and Programming*, pages 273–284. Springer, 2001.

[34] Deeparnab Chakrabarty and Chaitanya Swamy. Approximation algorithms for minimum norm and ordered optimization problems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC'19)*, pages 126–137, 2019.

[35] Parinya Chalermsook and Julia Chuzhoy. Maximum independent set of rectangles. In *Proceedings of the twentieth annual ACM-SIAM symposium on discrete algorithms*, pages 892–901. SIAM, 2009.

[36] Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-eth to fpt-inapproximability: Clique, dominating set, and more. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 743–754. IEEE, 2017.

[37] Parinya Chalermsook, Ameet Gadekar, Kamyar Khodamoradi, and Joachim Spoerhase. Independent set in k-claw-free graphs: Conditional χ-boundedness and the power of lp/sdp relaxations, 2023.

[38] Parinya Chalermsook and Daniel Vaz. A note on fractional coloring and the integrality gap of LP for maximum weight independent set. *Electronic Notes in Discrete Mathematics*, 55:113–116, 2016.

[39] Parinya Chalermsook and Bartosz Walczak. Coloring and maximum weight independent set of rectangles. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 860–868. SIAM, 2021.

[40] Timothy M Chan and Sariel Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 333–340, 2009.

[41] Yuk Hei Chan and Lap Chi Lau. *On Linear and Semidefinite Programming Relaxations for Hypergraph Matching*, pages 1500–1511.

[42] Yuk Hei Chan and Lap Chi Lau. On linear and semidefinite programming relaxations for hypergraph matching. *Mathematical programming*, 135(1-2):123–148, 2012.

[43] Barun Chandra and Magnús M Halldórsson. Greedy local improvement and weighted set packing approximation. *Journal of Algorithms*, 39(2):223–240, 2001.

[44] Frédéric Chataigner, G Manić, Yoshiko Wakabayashi, and Raphael Yuster. Approximation algorithms and hardness results for the clique packing problem. *Discrete Applied Mathematics*, 157(7):1396–1406, 2009.

[45] Xingyu Chen, Brandon Fain, Liang Lyu, and Kamesh Munagala. Proportionally fair clustering. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, volume 97, pages 1032–1041. PMLR, 2019.

[46] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (NIPS'17)*, pages 5029–5037, 2017.

[47] Eden Chlamtáč, Yury Makarychev, and Ali Vakilian. Approximating fair clustering with cascaded norm objectives. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*, pages 2664–2683. SIAM, 2022.

[48] Huairui Chu. A tight lower bound for compact set packing, 2023.

[49] Maria Chudnovsky and Paul Seymour. Claw-free graphs vi. colouring. *Journal of Combinatorial Theory, Series B*, 100(6):560–572, 2010.

[50] Vincent Cohen-Addad. Approximation schemes for capacitated clustering in doubling metrics. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, pages 2241–2259. SIAM, 2020.

[51] Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT approximations for k-median and k-means. In *Proc. 46th International Colloquium on Automata, Languages, and Programming (ICALP'19)*, volume 132 of *LIPIcs*, pages 42:1–42:14, 2019.

[52] Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT Approximations for k-Median and k-Means. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 42:1–42:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[53] Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT Approximations for k-Median and k-Means. In *ICALP*, volume 132, pages 42:1–42:14. Dagstuhl, 2019.

[54] Vincent Cohen-Addad, C. S. Karthik, and Euiwoong Lee. *On Approximability of Clustering Problems Without Candidate Centers*, pages 2635–2648.

[55] Vincent Cohen-Addad and CS Karthik. Inapproximability of clustering in lp metrics. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS'19)*, pages 519–539. IEEE, 2019.

[56] Vincent Cohen-Addad, Kasper Green Larsen, David Saulpic, and Chris Schwiegelshohn. Towards optimal lower bounds for k-median and k-means coresets. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC'22)*, pages 1038–1051, 2022.

[57] Vincent Cohen-Addad, Marcin Pilipczuk, and Michal Pilipczuk. Efficient approximation schemes for uniform-cost clustering problems in planar graphs. In *27th Annual European Symposium on Algorithms (ESA'19)*, volume 144, pages 33–1. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[58] Vincent Cohen-Addad, David Saulpic, and Chris Schwiegelshohn. A new coreset framework for clustering. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC'21)*, pages 169–182, 2021.

[59] Marek Cygan. Improved approximation for 3-dimensional matching via bounded pathwidth local search. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 509–518. IEEE, 2013.

[60] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer Publishing Company, Incorporated, 1st edition, 2015.

[61] H. Gökalp Demirci and Shi Li. Constant approximation for capacitated k-median with $(1+\epsilon)$-capacity violation. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP'16)*, volume 55 of *LIPIcs*, pages 73:1–73:14, 2016.

[62] Hu Ding and Jinhui Xu. A unified framework for clustering constrained data without locality property. *Algorithmica*, 82(4):808–852, 2020.

[63] Rodney G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer Publishing Company, Incorporated, 2012.

[64] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *STOC*, page 569–578. ACM, 2011.

[65] Noah Fleming, Pravesh Kothari, Toniann Pitassi, et al. Semialgebraic proofs and efficient algorithm design. *Foundations and Trends® in Theoretical Computer Science*, 14(1-2):1–221, 2019.

[66] J. Flum and M. Grohe. *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[67] Eli Fox-Epstein, Philip N Klein, and Aaron Schild. Embedding planar graphs into low-treewidth graphs with applications to efficient approximation schemes for metric problems. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)*, pages 1069–1088. SIAM, 2019.

[68] Zachary Friggstad and Yifeng Zhang. Tight analysis of a multiple-swap heuristic for budgeted red-blue median. *arXiv preprint arXiv:1603.00973*, 2016.

[69] Ameet Gadekar. On the parameterized complexity of compact set packing. In Chun-Cheng Lin, Bertrand M. T. Lin, and Giuseppe Liotta, editors, *WALCOM: Algorithms and Computation*, pages 359–370, Cham, 2023. Springer Nature Switzerland.

[70] Mehrdad Ghadiri, Samira Samadi, and Santosh Vempala. Socially fair k-means clustering. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 438–448, 2021.

[71] Mehrdad Ghadiri, Mohit Singh, and Santosh S Vempala. Constant-factor approximation algorithms for socially fair k-clustering. *arXiv preprint arXiv:2206.11210*, 2022.

[72] Dishant Goyal and Ragesh Jaiswal. Tight fpt approximation for socially fair clustering. *Information Processing Letters*, 182:106383, 2023.

[73] Dishant Goyal and Ragesh Jaiswal. Tight fpt approximation for socially fair clustering. *Information Processing Letters*, page 106383, 2023.

[74] Martin Grötschel, László Lovász, and Alexander Schrijver. Polynomial algorithms for perfect graphs. In *North-Holland mathematics studies*, volume 88, pages 325–356. Elsevier, 1984.

[75] Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '98, page 649–657, USA, 1998. Society for Industrial and Applied Mathematics.

[76] A. Gupta, R. Krauthgamer, and J.R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 534–543, 2003.

[77] M Hajiaghayi, Rohit Khandekar, and Guy Kortsarz. Local search algorithms for the red-blue median problem. *Algorithmica*, 63(4):795–814, 2012.

[78] Magnús M Halldórsson, Jan Kratochvıl, and Jan Arne Telle. Independent sets with domination constraints. *Discrete Applied Mathematics*, 99(1-3):39–54, 2000.

[79] Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM Symposium on Theory of computing (STOC'04)*, pages 291–300, 2004.

[80] Refael Hassin and Shlomi Rubinstein. An approximation algorithm for maximum triangle packing. *Discrete Applied Mathematics*, 154(6):971–979, 2006.

[81] J. Hastad. Clique is hard to approximate within n1-. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, FOCS '96, page 627, USA, 1996. IEEE Computer Society.

[82] Elad Hazan, Shmuel Safra, and Oded Schwartz. On the complexity of approximating k-set packing. *computational complexity*, 15(1):20–39, 2006.

[83] Dorit S Hochbaum. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Applied Mathematics*, 6(3):243–254, 1983.

[84] Lingxiao Huang and Nisheeth K Vishnoi. Coresets for clustering in euclidean spaces: importance sampling is nearly optimal. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC'20)*, pages 1416–1429, 2020.

[85] Mark Jones, Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Ondřej Suchý. Parameterized complexity of directed steiner tree on sparse graphs. In Hans L. Bodlaender and Giuseppe F. Italiano, editors, *Algorithms – ESA 2013*, pages 671–682, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[86] Ioannis Katsikarelis, Michael Lampis, and Vangelis Th Paschos. Structural parameters, tight bounds, and approximation for (k, r)-center. *Discrete Applied Mathematics*, 264:90–117, 2019.

[87] Jeong Han Kim. The Ramsey number $R(3,t)$ has order of magnitude $t^2/\log t$. *Random Struct. Algorithms*, 7(3):173–207, oct 1995.

[88] Ravishankar Krishnaswamy, Amit Kumar, Viswanath Nagarajan, Yogish Sabharwal, and Barna Saha. The matroid median problem. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1117–1130. SIAM, 2011.

[89] Ravishankar Krishnaswamy, Shi Li, and Sai Sandeep. Constant approximation for k-median and k-means with outliers via iterative rounding. In *STOC*, pages 646–659, 2018.

[90] Amit Kumar, Yogish Sabharwal, and Sandeep Sen. Linear-time approximation schemes for clustering problems in any dimensions. *J.ACM*, 57(2):1–32, 2010.

[91] Piyush Kumar and E. Alper Yildirim. An algorithm and a core set result for the weighted euclidean one-center problem. *INFORMS Journal on Computing*, 21(4):614–629, 2009.

[92] G. Laporte, S. Nickel, and F. S. da Gama. *Location Science*. Springer, 2015.

[93] Jean B Lasserre. An explicit exact sdp relaxation for nonlinear 0-1 programs. In *Integer Programming and Combinatorial Optimization: 8th International IPCO Conference Utrecht, The Netherlands, June 13–15, 2001 Proceedings 8*, pages 293–303. Springer, 2001.

[94] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.

[95] Monique Laurent. A comparison of the sherali-adams, lovász-schrijver, and lasserre relaxations for 0–1 programming. *Mathematics of Operations Research*, 28(3):470–496, 2003.

[96] Liane Lewin-Eytan, Joseph Seffi Naor, and Ariel Orda. Routing and admission control in networks with advance reservations. In *Approximation Algorithms for Combinatorial Optimization: 5th International Workshop, APPROX 2002 Rome, Italy, September 17–21, 2002 Proceedings 5*, pages 215–228. Springer, 2002.

[97] Jian Li, Ke Yi, and Qin Zhang. Clustering with diversity. In *Automata, Languages and Programming, 37th International Colloquium (ICALP'19)*, volume 6198, pages 188–200. Springer, 2010.

[98] Inge Li Gørtz and Anthony Wirth. Asymmetry in k-center variants. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX'03)*, pages 59–70. Springer, 2003.

[99] Bingkai Lin. A Simple Gap-Producing Reduction for the Parameterized Set Cover Problem. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 81:1–81:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[100] Bingkai Lin. Constant approximating k-clique is w[1]-hard. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1749–1756. ACM, 2021.

[101] Bingkai Lin, Xuandi Ren, Yican Sun, and Xiuhan Wang. *Constant Approximating Parameterized <italic>k</italic>-SETCOVER is W[2]-hard*, pages 3305–3316.

[102] László Lovász. A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B*, 13(2):95–98, 1972.

[103] Yury Makarychev and Ali Vakilian. Approximation algorithms for socially fair clustering. In *Conference on Learning Theory (COLT'21)*, pages 3246–3264. PMLR, 2021.

[104] Dániel Marx. Can you beat treewidth? In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 169–179. IEEE, 2007.

[105] Jirı Matoušek. On approximate geometric k-clustering. *Discrete & Computational Geometry*, 24(1):61–84, 2000.

[106] Meike Neuwohner. An improved approximation algorithm for the maximum weight independent set problem in d-claw free graphs. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference)*, volume 187 of *LIPIcs*, pages 53:1–53:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[107] S. Nickel and J. Puerto. *Location Theory*. Springer Science & Business Media, 2005.

[108] Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. The effectiveness of Lloyd-type methods for the k-means problem. *J.ACM*, 59(6):1–22, 2013.

[109] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000.

[110] Pablo A Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96:293–320, 2003.

[111] Mihai Pătraşcu and Ryan Williams. On the possibility of faster sat algorithms. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1065–1075. SIAM, 2010.

[112] Ján Plesník. A heuristic for the p-center problems in graphs. *Discrete Applied Mathematics*, 17(3):263–268, 1987.

[113] Thomas Rothvoß. The lasserre hierarchy in approximation algorithms. *Lecture Notes for the MAPSP*, pages 1–25, 2013.

[114] Karthik C. S. and Subhash Khot. Almost polynomial factor inapproximability for parameterized k-clique, 2021.

[115] Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 1283–1296, New York, NY, USA, 2018. Association for Computing Machinery.

[116] Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. *J. ACM*, 66(5), aug 2019.

[117] Alex Scott and Paul Seymour. A survey of χ-boundedness. *Journal of Graph Theory*, 95(3):473–504, 2020.

[118] Hanif D Sherali and Warren P Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.

[119] Amnon Ta-Shma. Explicit, almost optimal, ϵ-balanced codes. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proc. 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC'17)*, pages 238–251. ACM, 2017.

[120] Arie Tamir. The k-centrum multi-facility location problem. *Discrete Applied Mathematics*, 109(3):293–307, 2001.

[121] Suhas Thejaswi, Ameet Gadekar, Bruno Ordozgoiti, and Michal Osadnik. Clustering with fair-center representation: Parameterized approximation algorithms and heuristics. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 1749–1759, New York, NY, USA, 2022. Association for Computing Machinery.

[122] Suhas Thejaswi, Bruno Ordozgoiti, and Aristides Gionis. Diversity-aware k-median: Clustering with fair center representation. In *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference (ECML PKDD'21), Bilbao, Spain, September 13-17, 2021, Proceedings, Part II*, volume 12976, pages 765–780. Springer, 2021.

[123] Suhas Thejaswi, Bruno Ordozgoiti, and Aristides Gionis. Diversity-aware k-median: Clustering with fair center representation. In *ECML-PKDD*, pages 765–780. Springer, 2021.

[124] Theophile Thiery and Justin Ward. An improved approximation for maximum weighted k-set packing. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 1138–1162. SIAM, 2023.

[125] Andrea Vattani. The hardness of k-means clustering in the plane. 2010.

[126] Vijay V. Vazirani. *Approximation Algorithms*. Springer Publishing Company, Incorporated, 2010.

[127] RR Vemuganti. Applications of set covering, set packing and set partitioning models: A survey. In *Handbook of combinatorial optimization*, pages 573–746. Springer, 1998.

[128] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, USA, 1st edition, 2011.

[129] JED YANG. On coloring claw-free graphs, 2007.

Aalto-DT 227/2023

This thesis explores the intricate domains of clustering and graph packing, spanning optimization, machine learning, data mining, computational geometry, and operations research. Addressing NP-hard problems, it adopts the dual lenses of approximation algorithms and parameterized algorithms, with an emphasis on the emerging paradigm of parameterized approximation algorithms.

The thesis unfolds in two distinct parts. In Part I, the focus is on designing parameterized approximation algorithms for various clustering problems, pushing the boundaries of existing knowledge. Introducing a novel concept of *Scatter Dimension*, the thesis presents an *EPAS* framework for several clustering problems, encompassing classical problems like *k-median, k-means*, and *k-center*, across several metric spaces, unifying previously disparate results, and providing new results for advanced objectives that incorporate fairness and robustness. The thesis also tackles other clustering problems, including *Robust clustering* and *Diversity-aware clustering*, presenting tight parameterized approximation algorithms.

Part II complements the first part, focusing on lower bounds for graph packing problems. The thesis presents a parameterized dichotomy for the notoriously hard *Set Packing* problem. It also explores the intricate connection between approximating the maximum independent set problem in *k-claw-free graphs* and various convex relaxations.

9 789526 416021

ISBN 978-952-64-1602-1 (printed)
ISBN 978-952-64-1603-8 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Computer Science
www.aalto.fi