

# Fixed Parameter Tractable Algorithm and Coreset for the Ordered $k$ -Median problem

Michał Osadnik

## School of Science

Thesis submitted for examination for the degree of Master of  
Science in Technology.

Espoo 17.02.2023

## Supervisor

Prof. Parinya Chalermsook

## Advisors

Dr Kamyar Khodamoradi

Dr Joachim Spoerhase

Copyright © 2023 Michał Osadnik



---

**Author** Michał Osadnik

---

**Title** Fixed Parameter Tractable Algorithm and Coreset for the Ordered  $k$ -Median problem

---

**Degree programme** Computer, Communication and Information Sciences

---

**Major** Computer Science

---

**Code of major** SCI3042

---

**Supervisor** Prof. Parinya Chalermsook

---

**Advisors** Dr Kamyar Khodamoradi, Dr Joachim Spoerhase

---

**Date** 17.02.2023

---

**Number of pages** 66

---

**Language** English

---

### Abstract

Clustering problems are among the most central problems that arise in many research areas, such as machine learning, data mining, and algorithms. The goal of such problems is to partition data points into  $k$  clusters to optimize certain objectives. Classical clustering objectives ( $k$ -MEANS,  $k$ -MEDIAN,  $\ell$ -CENTRUM) have received significant attention for decades from researchers and practitioners. More recent applications require us to handle much more complex clustering objectives and constraints (e.g., fairness, diversity, and ordering), which are much less understood. This thesis presents new algorithmic results for the ORDERED  $k$ -MEDIAN problem, a generalization of  $k$ -MEDIAN and  $\ell$ -CENTRUM.

Informally, ORDERED  $k$ -MEDIAN addresses the problem of finding the optimal set of centers minimizing the weighted sum of distances of clients to the closest center. A weight vector is introduced to make the solution balanced. Effectively, clients served with higher connection costs contribute with larger weights.

Apart from designing the algorithm for solving the problem, our interest points toward *coresets*. Coreset is a highly-applicable data-reduction technique, transforming points into a small (weighted) points set, such that for every set of potential centers, the objective value is approximated within  $1 \pm \epsilon$  factor.

**Algorithm** We introduce the algorithm for ORDERED  $k$ -MEDIAN, which yields a constant approximation ratio. Our ratio is significantly better than known in the literature [1, 2]. Also, this thesis proposes a novel approach to analyze the ratio  $t$  between the smallest and the most significant weight in the weight vector.

We propose  $\min(1 + 2/(1 + t), (e + 2)/(t \cdot e)) + \epsilon$  approximation algorithm running in fixed-parameters tractable algorithm (ignoring the impact of some variables) for problem restricted to  $\mathbf{L}_p$  norm. Without considering parameter  $t$ , the algorithm yields  $3 + \epsilon$  approximation factor. The crucial element of the algorithm is the coreset construction.

**Coreset** The coresets for  $k$ -MEDIAN (less general problem than ORDERED  $k$ -MEDIAN) are already well-studied. The recent results almost match the size lower bounds for both Euclidean  $\Omega(\frac{k}{\epsilon^2})$  and general metric space  $\Omega(\frac{k \log n}{\epsilon})$  [3], where  $n$  is a size of the data set. Notably, the size does not become exponentially dependent on the number of dimensions. However, for ORDERED  $k$ -MEDIAN, the size needs to

be at least  $\Omega(\frac{k}{\epsilon^d})$  [4]. The only known results in this area introduced coresets of size  $\mathcal{O}(\frac{k^2 \log^2 n}{\epsilon^{d+3}})$  for Euclidean space with  $\mathbf{L}_2$  norm [5].

Extending the previous work, we propose a strong and simultaneous (correct for all vectors) coresets construction for ORDERED  $k$ -MEDIAN. We present results for the Euclidean space with the  $\mathbf{L}_p$  norm, which matches the previous construction size for  $\mathbf{L}_2$ . Moreover, we analyze the impact of restringing the weights vector with the lower bound for the ratio between the smallest and most significant factors. That relaxed versatility allows for parameterizing the coresets size and smooth interpolation between  $k$ -MEDIAN and unrestricted ORDERED  $k$ -MEDIAN. Our results serve as a vital part in a fixed-parameter tractable algorithm for the Euclidean ORDERED  $k$ -MEDIAN problem.

Eventually, the hardness analysis concludes the thesis. We propose further research direction and highlight the limitations we encounter.

---

**Keywords** approximation algorithms, fixed-parameter tractability,  $k$ -median, ordered  $k$ -median, clustering, core-sets, hardness

---

## Acknowledgements

I would like to express profound gratitude to everyone involved in the process of writing this thesis and supporting me on the academic journey. With the help of many, I am able to accomplish this process.

First, I need to thank my supervisor, prof. Parinya Chalermsook, for the opportunity to work in the research group. His guidance and given the freedom to explore multiple projects significantly stimulated my growth and allowed for a deep engagement with the topic.

My supervisors, dr Kamyar Khodamoradi and dr Joachim Spoerhase supported the process with thoughtful reviews. Their suggestions served as an inspiration and motivation in my work. Also, I am thankful for them pointing me to this research area and showing potential directions.

Then, I want to thank the whole research group and the entire Aalto University community for creating a valuable and inspiring environment. Apart from colleagues, I thank all teachers for the fascinating courses I had the pleasure to participate.

Also, I would like to highlight the importance of my previous university, AGH University of Science and Technology, in Kraków. Mainly, I need to mention the contribution of prof. Piotr Faliszewski. He excited me with theoretical computer science and computational complexity.

Last but not least, thanks to everyone helping me personally: parents, family, friends, and my partner. Writing this thesis was demanding not only scientifically, so there are no words to express how generous the emotional assistance I received was.

# Contents

<b>Abstract</b>	<b>3</b>
<b>Contents</b>	<b>6</b>
<b>1 Introduction</b>	<b>8</b>
1.1 Current research	8
1.2 Our contribution	9
1.3 Thesis structure	10
<b>2 Preliminaries</b>	<b>11</b>
2.1 Clustering problems overview	11
2.1.1 Metric $k$ -MEDIAN as a base for further problems	11
2.1.2 Objective function	12
2.1.3 Fairness	13
2.2 ORDERED $k$ -MEDIAN	14
<b>3 Strong simultaneous coresets for Ordered <math>k</math>-Median in Euclidean space with <math>L_p</math> norm</b>	<b>16</b>
3.1 Introduction	16
3.2 Overview of the construction	18
3.3 Non-simultaneous coresets for $\ell$ -CENTRUM with clients defined on lines	20
3.4 Non-simultaneous coresets for $\ell$ -CENTRUM	30
3.5 Coresets for $k$ -MEDIAN	32
3.6 Simultaneous coresets for ORDERED $k$ -MEDIAN	32
3.7 Simultaneous coresets for $\ell$ -CENTRUM	35
<b>4 Fixed-Parameter Tractable approximation algorithm for Ordered <math>k</math>-Median</b>	<b>36</b>
4.1 Preliminaries	36
4.2 Overview on the construction	36
4.3 Client set reduction	37
4.4 $(3 + \mathcal{O}(\epsilon))$ -approximation by finding <i>leaders</i> and <i>radii</i>	37
4.5 $(1 + \frac{2}{1+t} + \mathcal{O}(\epsilon))$ -approximation by submodular maximization	40
4.5.1 A pproximately submodular function subject to the partition matroid and cardinality constraints	40
4.5.2 Modified algorithm and approximately submodularity factor function analysis	42
4.6 A naive approach for large $t$	47
4.7 Summary	48
<b>5 Hardness of <math>\ell</math>-centrum and Ordered <math>k</math>-Median approximation</b>	<b>49</b>
5.1 Hardness of $k$ -CENTRUM approximation	49
5.2 The hardness of $\ell$ -CENTRUM approximation with respect to $\ell$	50
5.3 The hardness of ORDERED $k$ -MEDIAN approximation with respect to the parameter $t$	53

<b>6</b>	<b>Conclusions</b>	<b>57</b>
6.1	Summary . . . . .	57
6.2	Further work . . . . .	57

# 1 Introduction

Clustering is a comprehensive set of problems typically aiming to find similarities between nodes in the given input and grouping them into partitions that the representative can represent. In other words, given the set of clients, and facilities (that might be discrete, continuous, or equal to the client set), we seek to find  $k$  facilities minimizing the cost of assigning each client to the closest open facility. This group of problems applies to several theoretical and practical problems. A good example fitting modern times is locating proxy servers in the web [6]. The most famous problem that is handy in explaining is opening retail facilities to make them reachable by customers. Many problems' variations are applicable in Machine Learning and Data Science. In those cases, research is related chiefly to Euclidean Space (but not necessarily  $\mathbf{L}_2$  norm) and connected with *node embedding*, namely, the transformation of nodes of a knowledge graph into vectors set (considered as points in  $\mathbb{R}^d$ ) [7].

Therefore, clustering problems tend to answer important multi-disciplinary questions and are used to model various real-life situations. However, answering real-life situations bring together real-life limitations. Particularly, one may ask, what does it mean that the specific problem solution is *fair*? How can we enforce additional constraints to control the *fairness* of the solution? Also, how can we act against real-life problems (e.g., social inequalities or discrimination) by providing additional data (e.g., description of protected groups) or modifying the objective function? Those questions are an essential area of interest for modern research in clustering, e.g., [8, 9, 10, 11, 12, 13, 14].

Those motivations evolved into the formulation of the ORDERED  $k$ -MEDIAN problem. Although the problem instance itself does not explicitly expect any input containing a description of protected groups, it includes an additional weight vector used to emphasize the impact of the clients with the highest cost incurred (being in the least favorable position). The solution to the problem is the selection of facilities, such that the weighted cost of the clients' assignment to the closest facilities is minimized. Similar motivation served for similar problems, like  $k$ -CENTRUM,  $\ell$ -CENTRUM or  $k$ -CENTDIAN. ORDERED  $k$ -MEDIAN is a generalization of them.

Our research focuses on *fixed-parameter tractable algorithms*, an active algorithmic framework that aims for approximation algorithms with the running time possibly super-polynomial with respect to some variables. That direction is a subject of active research in the context of clustering algorithms [15, 13, 16, 17, 18].

## 1.1 Current research

The generality of ORDERED  $k$ -MEDIAN renders it intriguing from the perspective of possible approaches [19]. Particularly, techniques working efficiently for other problems, being special cases of ORDERED  $k$ -MEDIAN, fail for the general problem instance [20].

Due to the universality of the problem, many algorithms have been designed with the limitations of various constraints. E.g., polynomial-time algorithms were obtained

for locating a single facility [21, 22], where the metric space is induced by a network. Also, multi-facility problems have been solved efficiently under additional structures, particularly by combining tree metrics and specific forms of penalty weights [23, 24].

The first provably-good approximation algorithm for the ORDERED  $k$ -MEDIAN problem was devised in 2019 by Aouad, and Segev [19]. Recently, Byrka et al. [2] and Chakrabarty and Swamy [1] independently presented the constant-factor approximation algorithm for the ORDERED  $k$ -MEDIAN problem, obtaining the approximation ratios of 38 and  $18 + \epsilon$ . To the best of our knowledge, there is no research for the ORDERED  $k$ -MEDIAN problem restricted to Euclidean space. For a more comprehensive overview, we refer to the literature [21, 25].

The open questions, being an area of our interest, are as follows:

1. Can we devise better algorithms with *fixed-parameter tractable algorithms*?
2. Can we design algorithms limiting our focus to the  $\mathbf{L}_p$  metric?
3. Can we propose a different, yet unexplored, way to limit the class of general ORDERED  $k$ -MEDIAN instances?
4. Can we say something about the hardness of ORDERED  $k$ -MEDIAN?

## 1.2 Our contribution

We answer affirmatively to all of those questions in the following thesis.

The novel idea introduced in this thesis regarding approaching the ORDERED  $k$ -MEDIAN problem is tracking the ratio between the smallest and most significant factor in weights vector  $v$ . That approach was inspired by [26] for the weighted  $\ell$ -CENTRUM problem. We denote this parameter as  $t$ .

$$t = \frac{v_n}{v_1}.$$

However, for the sake of simplicity, we can assume  $v_1 = 1$  and scale other weights. Then,

$$t = v_n.$$

We can assume that  $t \leq 1$ . In the general case of ORDERED  $k$ -MEDIAN,  $t$  is unrestricted. However, while assuming a lower bound for  $t$ , we obtain a modified formulation of the problem that yields promising results from the perspective of determining coresets and designing a *fixed-parameter tractable* (FPT) algorithm<sup>1</sup>. Intuitively, this parameter allows for smooth interpolation between  $k$ -MEDIAN and unrestricted ORDERED  $k$ -MEDIAN. When  $t = 0$ , we have the general well-established ORDERED  $k$ -MEDIAN, and where  $t = 1$ , we observe the instance of  $k$ -MEDIAN. Figure 3 depicts the example of a weights vector restricted by parameter  $t$ .

---

<sup>1</sup>FPT( $k$ ) is an algorithm with the running time  $f(k) \cdot \text{poly}(n)$  for some function  $f$

Our primary focus was to design the FPT algorithm inspired by [15] with respect to the parameter  $t$ . We managed to obtain the following approximation ratio for Euclidean space with  $\mathbf{L}_p$  norm:

$$\min\left(1 + \frac{2}{1+t}, \frac{e+2}{t \cdot e}\right) + \mathcal{O}(\epsilon).$$

Specifically,

- If  $t = 1$ , we recover  $k$ -MEDIAN and obtain ratio  $1 + \frac{2}{e} + \mathcal{O}(\epsilon)$ .
- If  $t = 0$ , we consider unrestricted case and obtain ratio  $1 + \frac{2}{e} + \mathcal{O}(\epsilon)$ .

Those ratios in the general metric would be tight due to the hardness analysis in Section 5.

During the design of the algorithm, we noticed that the coresets construction available in literature [5] (the precise definition of coresets is contained in the Section 3) limits us the  $\mathbf{L}_2$  norm. We did not manage to find any other coresets construction available for the ORDERED  $k$ -MEDIAN problem. Therefore, inspired by [5], we decided to present our own construction. Although we did not manage to achieve results for the general metric space, we generalized the construction for any  $\mathbf{L}_p$  norm. Additionally, we analyzed the impact of restringing ORDERED  $k$ -MEDIAN instances by  $t$  on reducing the coresets size and the algorithm's running time.

Finally, we analyzed the polynomial hardness of the general (non-Euclidean) ORDERED  $k$ -MEDIAN problem, obtaining nontrivial results parametrized by  $t$ . Also, we introduce novel hardness results for the  $\ell$ -CENTRUM (and thus, ORDERED  $k$ -MEDIAN) problem, which matches known results for  $k$ -MEDIAN [27] as a special case.

### 1.3 Thesis structure

Our work contains six chapters. The first part of the thesis includes an introduction and the literature overview of the ORDERED  $k$ -MEDIAN problem. We position the problem among other research directions and highlight the motivation for the specific problem formulations. At the end of this part, we specify the contribution of the thesis and the novelties compared to the past work of other authors. We do not cover technical details here, since chapters may serve as independent work, and therefore, each of them contains its own introduction.

The second chapter covers preliminaries crucial to obtain a profound understanding of ORDERED  $k$ -MEDIAN and other, related problems.

The third chapter introduces a coresets construction, which is the extension of the algorithm known from literature into more general metric spaces together with a smooth parameterized interpolation between  $k$ -MEDIAN and ORDERED  $k$ -MEDIAN coresets.

Those results support the fourth chapter, presenting the complete approximation algorithm for the ORDERED  $k$ -MEDIAN problem with an analysis of the approximation ratio and running time.

Eventually, the fourth chapter analyzes the hardness of approximation.

The thesis ends with a summary and pointers for further research.

## 2 Preliminaries

### 2.1 Clustering problems overview

The clustering problems can be divided in many ways, intersecting and not exhaustive. In this overview, we ignore problems not located in the metric spaces, e.g., [28]. This introduction covers a few classifications known in the literature. Firstly, we will describe  $k$ -MEDIAN problem that is the most straightforward problem from the family. After that, the following subsection discusses different objectives' functions. Also, the formal definition of the ORDERED  $k$ -MEDIAN problem is given at this place. Additionally, the last part focuses on the notion of *fairness* that is not strictly related to the ORDERED  $k$ -MEDIAN problem but shares a similar motivation.

#### 2.1.1 Metric $k$ -Median as a base for further problems

The  $k$ -MEDIAN problem is one of the most well-studied clustering problems. As an input, we are given a set of points in a metric space (might be finite in the case of general metric space or continuous, e.g., the whole space in the case of vector spaces). The goal is to select  $k$  of these to be cluster centers and then assign each point to its closest selected center. If one point is assigned to some center, the cost incurred is proportional to the distance between this point and the center. The goal is to select the  $k$  centers that minimize the sum of the costs.

Namely, the instance is given as  $I = (X, F, M, k)$ , where  $M = (X \cup F, dist)$  is a metric space with a distance function  $dist$ . The cost for facility selection  $C \subseteq F$  is:

$$\text{cost}_{med}(X, C) = \sum_{x \in X} dist(x, C)$$

And,

$$\text{OPT} = \arg \min_{C \subseteq F, |C|=k} \text{cost}_{med}(X, C)$$

For simplicity,  $dist(x, F)$  denotes the distance to the closest facility from the set  $C$ . Particularly,

$$dist(x, F) = \min_{f \in F} dist(x, f)$$

Currently, the best-known results for a general metric space yield a 2.611 approximation ratio [29]. The best lower bound (that we will prove again here with a different technique) is  $(1 + \frac{2}{e}) - \epsilon$  [27]. However, the recent approach being an inspiration for this thesis by Cohen et al. gives a tight  $(1 + \frac{2}{e} + \mathcal{O}(e))$  approximation algorithm in the fixed-parameter tractable time. Here, we consider fixing parameter  $k$  and then use the following definition.

$$\text{FPT}(k) = n^{\mathcal{O}(1)} \cdot f(k)$$

For the Euclidean variant (set of facilities being a metric space), techniques presented for general metrics often become ineffective. Any dependence on the facilities' set cardinality is problematic due to the continuity of the Euclidean space. However, there is a known 2.406 approximation algorithm for this variant [30].

### 2.1.2 Objective function

Although the most accessible formulation, the  $k$ -MEDIAN problem, is not often used in real data-based problems, this is not practically performing effectively in clustering problems. The reason is that the closest client's cost is incurred with the same weight as the cost of the most remote one. Effectively, using the retail store example again might lead to discriminating against individuals in the worst situation to increase the satisfaction of those already in a good position.

Therefore, this motivates the existence of another variant of the problem,  $k$ -MEANS. Historically, this variant is probably older (the term was used in 1967 [31]) and currently faces wider adaptation in algorithm designs. The only difference compared to  $k$ -MEDIAN within all variants is the objective function. The power of  $k$ -MEANS algorithm is due to its computational efficiency and the nature of ease at which it can be used [32]. For  $k$ -MEANS,

$$\text{cost}_{\text{means}}(X, C) = \sum_{x \in X} \text{dist}(x, C)^2$$

Due to the nature of applications, the problem is typically considered in Euclidean space. Also, in Euclidean space, the distance becomes simply the sum of the squared distances across all dimensions. Non-euclidean variants are not subject to that extensive research and even consider as a different problem variant [33].

The best known polynomial-time approximation ratios for the general metric are 6.357 [34] and 3.943, considering FPT( $k$ ) running time [15]. The best known polynomial-time approximation ratio in Euclidean space is 5.912 [30] and  $1 + \epsilon$  with fixing  $k$ . [35]

Figure 1 depicts the differences in  $k$ -MEANS and  $k$ -MEDIAN highlighting the weakness of the  $k$ -MEDIAN formulation.

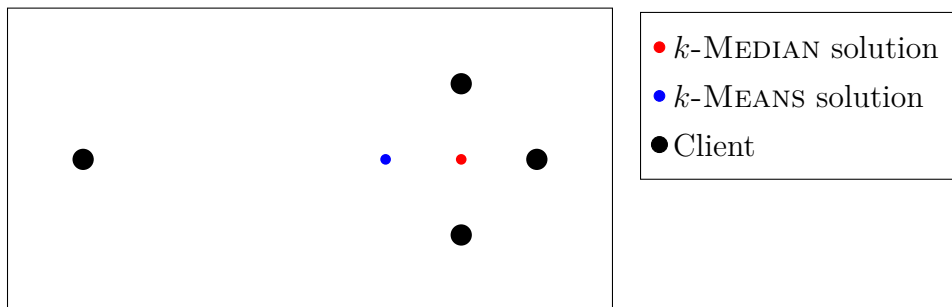


Figure 1: Example of the different optimal facility location depending on the used objective function in the complete Euclidean space. Furthermore, this is the complete problem optimal solution for  $k = 1$ .

$\ell$ -CENTRUM (using notion by Chakrabarty and Swamy [36] also known as  $p$ -FACILITY  $k$ -CENTRUM [24] or simply  $p$ -CENTRUM [5], which is significantly misleading) represents another approach to model clustering objectives. Originally, this is a generalization of the well-researched problems:  $p$ -CENTRUM (that was initially formulated in 1970 [37] as  $m$ -CENTRUM, then generalized to the weighed case in [38, 39], and has an excellent overview in [40]) and  $k$ -MEDIAN.

$\ell$ -CENTRUM accepts two values –  $k$  and  $\ell$  representing the number of facilities to be selected and the number of the most further client (to the selected facilities set) included in a sum being the objective function of the problem instance.

Precisely,

$$\text{cost}_\ell(X, C) = \sum_{i=1}^p \max_{x \in X}^i \text{dist}(x, C)$$

Here,  $\max^i$  denotes the  $i$ -th greatest element.

[24] together with the problem formulation, presents a polynomial-time algorithms that solve the problem (optimally) on path and tree graphs. Hence, using tree-embedding gives a  $\mathcal{O}(\log n)$ -approximation for the general metric. Recently, two constant factor approximation has been derived. Byrka et al. presented a 15-approximation with LP rounding [29]. Chakrabarty and Swamy used a primal-dual approach instead, yielding a 7.5-approximation [1]. Interestingly, both of those research efforts considered  $\ell$ -CENTRUM as a special (and easier) case of ORDERED  $k$ -MEDIAN (Byrka called it RECTANGULAR ORDERED  $k$ -MEDIAN), which brings attention to the more general settings of the problem.

Figure 2 depicts the difference in results depending on the careful selection of  $\ell$  parameter.

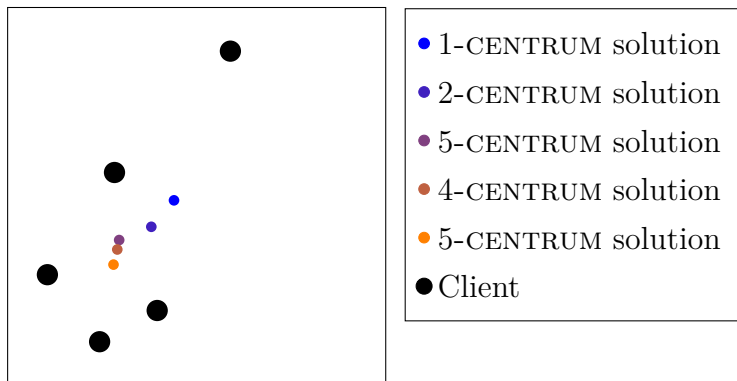


Figure 2: Example of the different optimal facility location for  $\ell$ -CENTRUM depending on the used value of  $\ell$  in the continuous Euclidean space. Furthermore, this is the complete problem optimal solution for  $k = 1$ .

### 2.1.3 Fairness

The additional view is *fairness* which has been a vital research topic recently. From the most apparent social perspective, it is essential to ensure that such algorithms are fair and are not biased towards or against specific groups in the population [9]. Oppositely, it is often intended to increase the impact of underrepresented individuals. Following the understanding of [8], the fairness idea is typically connected with imposing an additional requirement on top of an existing problem. That understanding was also transferred to clustering problems in [10] – apart from the objective function, we introduce other requirements specific to the input enforcing some constraints on the structure of the output.

Although the definition seems novel, some well-established problems can be classified into this category. Examples include CAPACITATED  $k$ -MEDIAN [41], SOCIALLY FAIR  $k$ -MEANS [42], BUDGETED RED-BLUE MEDIAN [43], MATROID MEDIAN [44], DIVERSITY-AWARE  $k$ -MEDIAN [11, 12, 13] or more generalized FAIR LABELED CLUSTERING [45].

This problem family serves as an inspiration for exploring the ORDERED  $k$ -MEDIAN problem being a convex combination of  $\ell$ -CENTRUM instances (and generalizing the  $\ell$ -CENTRUM problem). The additional weights vector, specific to the input, is required here and can be used to impose the *fairness* constraints. However, ORDERED  $k$ -MEDIAN does not fit perfectly into this definition of fairness. We do not explicitly define any protected classes and impose requirements on the output structure apart from minimizing the objective. Nevertheless, the additional flexibility of the objective function relates to the fairness of research to allow (fully controlled), ignoring the cost of the clients being already well-represented by selected facilities.

## 2.2 Ordered $k$ -Median

Eventually, this context is sufficient to formulate the ORDERED  $k$ -MEDIAN problem definition. As mentioned earlier, the instance is given as  $I = (X, F, M, k, v)$ , where  $M = (X \cup F, dist)$  is a metric space with a distance function  $dist$ . Compared to the definition of  $k$ -MEDIAN, we introduce a vector  $v$  of the size  $|X|$ . The elements of the vector are sorted in non-decreasing order. The solution's objective is obtained by sorting the distances to the nearest facilities in the non-decreasing order and then multiplying by the weights  $v$ . The most remote clients contribute more to the objective. Precisely, for  $C \subseteq F$ :

$$\text{cost}_v(X, C) = \sum_{i=1}^{|X|} v_i \cdot \max_{x \in X}^i dist(x, C).$$

That formulation generalizes problems:

1.  $k$ -MEDIAN.  $v_1 = v_2 = \dots v_n = 1$ .
2.  $\ell$ -CENTRUM (and  $k$ -CENTRUM as a special case)  $v_1 = \dots = v_\ell = 1$  and  $v_{\ell+1} = \dots = v_n = 0$ .
3.  $k$ -CENTDIAN as a convex combination of  $k$ -MEDIAN and  $\ell$ -CENTRUM objective [46].

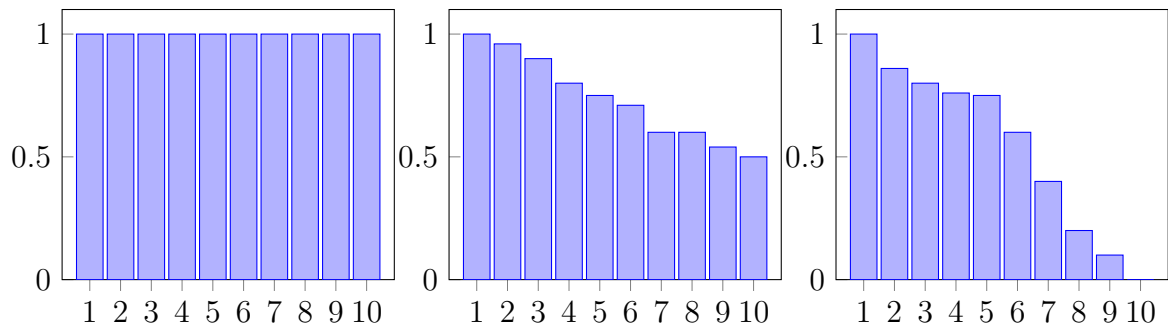


Figure 3: Interpolation between ORDERED  $k$ -MEDIAN and  $k$ -MEDIAN with  $n = 10$ . On the left side,  $t = 1$  (and the problem is matching  $k$ -MEDIAN objective). In the middle,  $t = 0.5$ . On the right side,  $t = 0$  (and the problem is unrestricted ORDERED  $k$ -MEDIAN)

### 3 Strong simultaneous coreset for Ordered $k$ -Median in Euclidean space with $L_p$ norm

#### 3.1 Introduction

The results presented here are crucial in the design of the algorithm in the next chapter, and this application motivates this effort. Coresets are a technique for data size reduction developed for a large family of problems in machine learning and statistics [47].

In the context of clustering, the central idea is to use the coreset construction as a step while designing the algorithm. The usage of coresets allows operating on a reduced client set, which significantly improves running time. Coreset properties are size and the approximation ratio. Typically, we refer to  $\epsilon$ -coresets considering the approximation ratio and aim to obtain the smallest size. Precisely, we focus on a coreset construction, where the input set  $X$  is transformed into set  $D$  such that:

$$\text{cost}(D, C) \in (1 \pm \epsilon) \cdot \text{cost}(X, C).$$

Here, we consider a cost without specifying the clustering objective because the same definition applies to many clustering problems. Particularly, for ORDERED  $k$ -MEDIAN instance:

$$\text{cost}_v(D, C) \in (1 \pm \epsilon) \cdot \text{cost}_v(X, C).$$

Coresets used for geometric approximation mainly live in low-dimensional spaces [48, 49] due to (possibly exponential) dependency on  $d$ . However, some of them might be extended to higher dimensions [50, 51]. There is coreset construction in general (non-Euclidean) metric space for some well-explored problems, e.g., [52] for  $k$ -MEDIAN.

In our deductions, we focus on the construction, where the coreset correctness property holds regardless of the selection of facilities (*strong* coreset [53]) and weight vector  $v$  (*simultaneous* coreset [54]). Nevertheless, the *simultaneity* is the desired property, the interesting question is how we can improve the coreset size (and construction time) by lowering coreset's versatility. Also, restricting simultaneity may serve as a tool for interpolating between the  $k$ -MEDIAN problem as a special case and the unrestricted ORDERED  $k$ -MEDIAN problem. For the algorithmic construction in the next chapter, we do not require the coreset to *be simultaneous*, but this remains a valuable property strengthening our results.

**Strong coresets.** Although for some applications [55, 56], this is enough if the coreset persists the objective cost (compared to the input client set) only for the optimal facilities selection, the majority of the application requires maintaining the cost for all proper (of the given size  $k$ ) selection of facilities. *Strong coresets* should preserve properties for all facilities selection. *Weak coresets* may maintain properties for a limited subset of queries (typically, optimal facility selection). Intuitively, strong coresets are a straightforward optimization for algorithms that are ineffective for the input client size. Therefore, strong coresets may serve as a weak coreset, but not the other way. Remarkably, there is no unifying agreement on the accuracy guarantee of weak coresets, and conflicts occur in the literature [57].

**Simultaneous coresets.** From the perspective of designing algorithms, it is valuable to introduce coresets that are not relying on the weights vector—especially considering ORDERED  $k$ -MEDIAN as a general case of  $k$ -MEDIAN,  $k$ -CENTRUM or  $\ell$ -CENTRUM, coresets for ORDERED  $k$ -MEDIAN become significantly attractive because the data reduction part of the algorithm might not *know* the clustering techniques used at the later stage.

This property has been described recently in [54] as *simultaneous* or *one-shot* coresets. From the perspective of  $\ell$ -CENTRUM problem, it means that knowledge of  $p$  is not needed for obtaining coreset. In ORDERED  $k$ -MEDIAN, the algorithms return a solution correct for all weight vectors.

Contrary to  $k$ -MEDIAN, the coresets problem for ORDERED  $k$ -MEDIAN is not yet thoroughly studied. The latest results introduced in [5], inspired by [58], yields  $\mathcal{O}_{\epsilon,d}(k \log^2 n)$ -size coresets computed in polynomial time<sup>2</sup>. The idea is limited only to Euclidean space with the  $\mathbf{L}_2$  norm.

The careful analysis presented below, together with a few modifications, shows that the procedure is effective for the Euclidean space also with a general  $\mathbf{L}_p$  norm for  $p \geq 1$ . Furthermore, tracking of the parameter  $t$  allows for a smooth interpolation between  $\mathcal{O}_{\epsilon,d}(k^2)$  size of  $k$ -MEDIAN coresets and  $\mathcal{O}_{\epsilon,d}(k^2 \log^2 n)$  size of ORDERED  $k$ -MEDIAN coresets.

Let  $\text{cost}_v$  denote the objective for ORDERED  $k$ -MEDIAN instance specified by the given weight vector  $v \in (0, 1)^n$ , where  $n$  is the size of the client set  $X \subset \mathbb{R}^d$ , and solution  $C$ .

$$\text{cost}_v(X, C) := \sum_{i \in \{1, \dots, |v|\}} v_i \cdot \max_{x \in X}^i \min_{x \in C} \text{dist}(x, c)$$

where  $\max^i$  is the  $i$ -th maximal element.

**Theorem 3.1** (Main Theorem). *For any value  $k \in \mathbb{Z}^+$ , clients  $X \subset \mathbb{R}^d$  of size  $|X| = n$ , any  $\epsilon \in (0, 1)$  and  $t \in [0, 1]$ , there is a weighted set  $D$  of size  $\mathcal{O}\left(\left(1 + \frac{(1-t) \log n}{\epsilon}\right)^2 \cdot \frac{k^2 \cdot d}{\epsilon^{d+1}}\right)$  such that:*

$$\text{cost}_v(D, C) \in (1 \pm \mathcal{O}(\epsilon)) \cdot \text{cost}_v(X, C)$$

*for all sets  $C \subset \mathbb{R}^d$  of size  $k$ , vector of weights  $v$  with parameter  $t$  (defined as in the introduction). Furthermore,  $D$  is computed in polynomial time.*

Before analyzing [Theorem 3.1](#), we highlight the motivation for that coreset size. Remarkably, the size seems highly technical and not attractive without following observations.

- If  $t = 1$ , then  $|D| \in \mathcal{O}\left(\frac{k^2 \cdot d}{\epsilon^{d+1}}\right) \in \mathcal{O}_{\epsilon,d}(k^2)$ . If  $t = 1$ , we obtain the instance of the  $k$ -MEDIAN problem. This size matches the coreset size for  $k$ -MEDIAN as defined in [Corollary 3.24](#).

---

<sup>2</sup> $\mathcal{O}_{\epsilon,d}$  notation indicates omitting the impact of  $d$  and  $\epsilon$ . We prefer to avoid this notation, but we use it here for simplicity and to continue the research direction of Braverman et al. [5]

- If  $t = 0$ , then  $|D| \in \mathcal{O}\left(\frac{\log^2 n \cdot k^2 \cdot d}{\epsilon^{d+3}}\right) \in \mathcal{O}_{\epsilon, d}(\log^2 n \cdot k^2)$ . We consider this case a general, unrestricted case of the ORDERED  $k$ -MEDIAN problem.

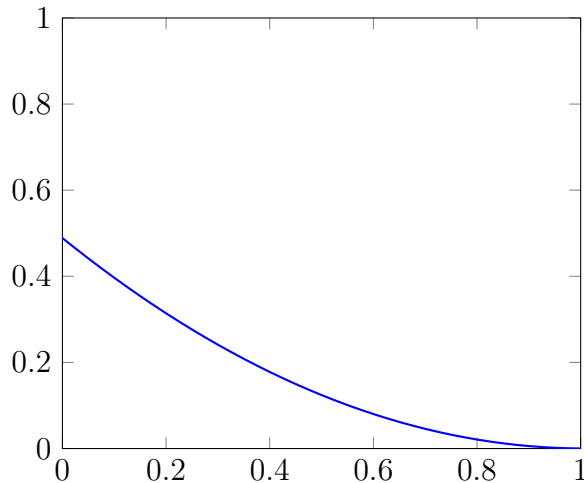


Figure 4: Interpolation of a coreset size while tracking parameter  $t$ . The  $\mathcal{O}(1)$  factor has been ignored and we assumed that  $\epsilon = 0.2$ ,  $d = 3$ ,  $n = 10^{11}$ ,  $k = 4$

Those results are close to results in [58] and [5] accordingly<sup>3</sup>. However, previous work was limited to  $\mathbf{L}_2$  norm. The only difference is the multiplicative factor  $d$ . However, this is a negligible factor compared to the fact that the previous results depend already on  $\epsilon^{-d-3}$ . The size of  $D$  is smoothly interpolating depending on  $t$  between the  $k$ -MEDIAN problem and the unrestricted ORDERED  $k$ -MEDIAN. Figure 4 provides an example of this fact.

The construction shares the main idea known from [5]. Nevertheless, contrary to that previous work, we prove a generalized theorem addressing Euclidean space with arbitrary  $\mathbf{L}_p$  norm and with the considering impact of  $t$ . Furthermore, our deductions are enriched with more precise mathematical analysis.

## 3.2 Overview of the construction

**Inspiration** Our construction is built on Braverman et al.’s [5] construction, so we start by explaining their ideas. We will highlight how our constructions differ from theirs.

Braverman et al.’s construction, inspired by [58], aims to find  $\mathcal{O}_{\epsilon, d}(\log^2 n \cdot k^2)$  representatives of the data set serving as a coreset. Representatives are devised in three steps.

The first step relies on the assumption that all clients lie on the collection of lines  $\mathcal{L}$ , given as input together with partitions of lines into subintervals. Braverman shows that sufficiently good representatives can be found by simply taking the mean

---

<sup>3</sup>Braverman et al. claim to obtain coreset of size  $\mathcal{O}\left(\frac{\log^2 n \cdot k^2}{\epsilon^{d+1}}\right)$ . However, the analysis of the bucketing procedure appears to be done incorrectly (ignoring the impact of the  $\epsilon$ ), and the size is  $\mathcal{O}\left(\frac{\log^2 n \cdot k^2}{\epsilon^{d+3}}\right)$ .

of each subinterval. In this step, the focus narrows to the  $\ell$ -CENTRUM problem with fixed  $\ell$  (*non-simultaneous* coresets).

The next step is to extend results into the general case, where clients are not restricted by lines set. Braverman shows, using  $\epsilon$ -nets, how to determine a *good* collection of lines and the proper projections to the lines-based case. In sum, each *mean point* represents all points projected into the interval.

Eventually, the last step introduces the procedure connecting *non-simultaneous*  $\ell$ -CENTRUM coresets, derived for carefully selected  $\ell$  values, into results for ORDERED  $k$ -MEDIAN.

In our work, we reuse that framework and present construction applicable to the more general metric. We could not use the majority of the original lemmas because their correctness proofs were limited to the  $\mathbf{L}_2$ . Therefore, we had to rely on norm properties and careful analysis instead of purely visual reasoning for  $\mathbf{L}_2$ . Additionally, the last section (*simultaneous* coresets) required a significant improvement due to the tracking of the  $t$  parameter.

Below, we overview three main parts of constructing the complete proof of [Theorem 3.1](#). We oversimplify some technicalities for the sake of improved presentation.

**$\ell$ -centrum coresets for lines-based case** The objective of that part is to design a non-simultaneous  $\ell$ -CENTRUM coresets for clients located on lines, which has an *additive* error  $\mathcal{O}_{\epsilon,d}(s + \sum_{j \in \mathcal{L}} t_j)$  ([Theorem 3.2](#)). The set of lines is denoted as  $\mathcal{L}$ . Variables  $s, t_j$  (for all  $j \in \mathcal{L}$ ) are given as a part of the input. Additionally, each line consists of sub-intervals, which are either *cheap* or *short*. The way lines are split into sub-intervals is also part of the problem instance. *Cheap* sub-intervals are defined as sub-intervals in which the total error that occurred while displacing the clients into the coresets point is small (specified by  $t_j$  for each line  $j \in \mathcal{L}$ ). *Short* sub-intervals are defined as those having a length specified by the value of  $s$ . The complete construction is to replace all points on each sub-interval with a single weighted point, which is a *mean* of all points. Noticeably, the construction does not influence the size of the coresets.

Instead of directly bounding the difference between the input and output set, we consider the intermediate set, defined to simplify the proof. Using that set, we only transform (replace by the *mean*) clients on sub-intervals fulfilling certain conditions ([Definition 3.3](#)). First, we bound the number of sub-intervals, breaking at least one of those conditions ([Lemma 3.4](#)). Then, we analyze the error obtained by transforming the sub-intervals, which fully meet the intermediate set requirements ([Lemma 3.12](#)). Those bounds are enough to conclude [Theorem 3.2](#).

**Projection procedure and sub-intervals' splitting** *Projection procedure* and *sub-intervals' splitting* allow the design of a non-simultaneous  $\epsilon$ -coresets for the  $\ell$ -CENTRUM problem by reusing the coresets for the lines-based case.

To determine the *projection centers*, the projection procedure employs any constant factor approximation algorithm for the  $\ell$ -CENTRUM problem. Then, those centers are covered with  $\epsilon$ -nets. From each projection center, we throw lines through every point from the  $\epsilon$ -net. Ultimately, we displace points from the client set to the

closest lines. Due to triangle inequality, this operation incorporates a sufficiently small multiplicative error ([Lemma 3.22](#)).

Then, to obtain the lines-based case, we need to specify not only the projection procedure but also the way sub-intervals are split. The greedy procedure is based on the sub-intervals' contribution to the approximate solution's cost. Eventually, the careful assignment of values  $s$  and  $t_j$  (for all  $j \in \mathcal{L}$ ) is crucial to obtain  $1 \pm \mathcal{O}(\epsilon)$  multiplicative factor of coresets.

Furthermore, due to  $\epsilon$ -nets properties and the greedy splitting analysis, we argue that the number of sub-intervals (and, therefore, the size of the coresets) is  $\mathcal{O}_{\epsilon,d}(k^2)$  ([Theorem 3.21](#)).

**Transforming  $\ell$ -centrum coresets into Ordered  $k$ -Median coresets** Eventually, the remaining part is to reuse non-simultaneous  $\ell$ -CENTRUM  $\epsilon$ -coresets to design the  $\epsilon$ -coresets for ORDERED  $k$ -MEDIAN (and simultaneous  $\epsilon$ -coresets for  $\ell$ -CENTRUM).

The strategy aims to represent all possible vectors (restricted by  $t$ ) as a convex combination of  $\ell$ -CENTRUM problems with fixed  $\ell$  value. We prove that we need at most  $\mathcal{O}(\log n)$  values of  $\ell$  (or less, if  $t$  is restricted) to cover all possible vectors within with  $1 \pm \epsilon$  approximation factor ([Lemma 3.26](#)). The algorithm for unionizing results for  $\ell$ -CENTRUM instances (for carefully selected values of  $\ell$ ) concludes the proof of the [Theorem 3.1](#)

### 3.3 Non-simultaneous coresets for $\ell$ -centrum with clients defined on lines

We present a construction of coresets together with a proof bounding error for any instance of  $\ell$ -CENTRUM, where all clients are located on lines given as input. We require client set  $X \in \mathbb{R}^d$  to lay on a set of lines  $\mathcal{L}$ . Furthermore, every line is built from sub-intervals satisfying specific requirements. This procedure requires previous knowledge about  $\ell$  that we consider as a part of the input (thus, non-simultaneous coresets). For the proof, we will use an intermediate set  $Z$  specific to the facilities selection. We will first bound the error between algorithm output  $D$  and intermediate set  $Z$ . Then, we will bound the error between the intermediate set  $Z$  and the instance client set  $X$ .

In this section, we will use parameters  $s$  and  $t_j$  (specific per line  $j$ ). Those parameters are defined carefully because eventually, in the next subsection, we will assign different values to them such that the sum  $s + \sum_{j \in \mathcal{L}} t_j$  will be  $\mathcal{O}(\epsilon) \cdot \text{OPT}$ . Before stating the lemma, we define the notation. The *coresets point* for interval  $Y$  and data set  $X$  is defined as:

$$\mu(Y) := \frac{1}{|Y \cap X|} \sum_{x \in X \cap Y} x$$

Let  $\delta(Y)$  denote the *cumulative error*:

$$\delta(Y) := \sum_{x \in X \cap Y} |x - \mu(Y)|$$

Additionally, distances' lengths are always specified in the given implicit norm. Namely,  $\text{dist}(a, b) = \text{len}(|ab|) = \|a, b\|_p$  for the given  $p$ . Particularly, intervals with the same distance in  $\mathbf{L}_2$  might not necessarily have the same distance in  $\mathbf{L}_p$  for  $p \neq 2$ .

**Theorem 3.2.** *Assume we are given values  $k, \ell \in \mathbb{Z}^+$ , a collection of lines  $\mathcal{L}$ , a set of clients  $X \subseteq \mathbb{R}^d$ , and values  $t_1, \dots, t_j, s > 0$  for  $j \in \mathcal{L}$  that satisfy the following properties:*

- *$X$  is partitioned into  $\{X_j \mid j \in \mathcal{L}, X_j \subseteq X\}$  and there are no clients outside the lines. We break ties arbitrarily.*
- *For  $j \in \mathcal{L}$ ,  $j$  is split into disjoint sub-intervals  $\mathcal{Y}_j$  such that  $\forall Y \in \mathcal{Y}_j$  covering all points:*
  - *Either,  $\text{len}(Y) < \mathcal{O}(\frac{\epsilon}{\ell} \cdot s)$  (interval is short)*
  - *Or  $\delta(Y) \leq \mathcal{O}(\frac{\epsilon}{k} \cdot t_j)$  (interval is cheap)*

*Define the coreset  $D$  as*

$$D = \{\mu(Y) \mid Y \in \mathcal{Y}_j, j \in \mathcal{L}\}$$

*Then, for all sets  $C \subset \mathbb{R}^d$  of size  $|C| = k$ :*

$$|\text{cost}_\ell(D, C) - \text{cost}_\ell(X, C)| \leq \mathcal{O}(\epsilon \cdot d) \cdot (s + \sum_{j \in \mathcal{L}} t_j)$$

The construction itself is obvious by the definition of set  $D$ . Therefore this is enough to prove the correctness of the bound. The proof contains two parts. Firstly, we define the intermediate set  $Z$  and bound the cost  $|\text{cost}_\ell(D, C) - \text{cost}_\ell(Z, C)|$ . Then, similarly, we bound  $|\text{cost}_\ell(Z, C) - \text{cost}_\ell(X, C)|$  for all  $k$ -subsets  $C \subset \mathbb{R}^d$ .

**Preliminaries** For  $x \in \mathbb{R}^d$ , let  $\xi(x)$  be the unique sub-interval that contains  $x$  obtained by the sub-interval split described in [Theorem 3.2](#). We say that  $\xi(x)$  is *induced* by  $x$ . Let  $\pi(x)$  be the *coreset point* of the given sub-interval, where the point is located, i.e.,  $\pi(x) = \mu(\xi(x))$ . We denote  $\Psi := \{x \mid \text{len}(\xi(x)) < \mathcal{O}(\frac{\epsilon}{\ell} \cdot s), x \in X\}$ . This set contains *short intervals*. Similarly, we define a complement  $\Gamma := X \setminus \Psi$  (*cheap intervals*). Let  $C'(c) := \{x \in X \mid \arg \max_{c' \in C} \text{dist}(x, c') = c\}$  denote a cluster induced by  $c \in C$ . It contains all clients  $x \in X$  whose closest neighbor in  $C$  is  $c'$ . Let  $\lambda(c, l) := \arg \min_{x \in l} \text{dist}(x, c)$  denote a *projection* of point  $c$  into line  $j \in \mathcal{L}$ . This is the closest point on line  $j$  (not necessarily in  $X$ ) to the center  $c$ . In  $\mathbf{L}_2$ , this is the point at the intersection of  $l$  and the line perpendicular to  $l$  that includes  $c$ . However, we cannot rely on this geometrical simplification in other norms. Let  $F_\ell(C, X) \subseteq X$  denote the subset of  $\ell$  farthest points in  $X$  from  $C$ .

**Definition 3.3** (Intermediate set  $Z$ ). *Set  $Z$  is defined by a transformation  $\gamma$  of set  $X = \{x_1, \dots, x_n\}$  into  $Z = \{z_1, \dots, z_n\} = \{\gamma(x_1), \dots, \gamma(x_n)\}$ . We define  $\gamma$  in the following way. If all conditions below are fulfilled:*

1. *Induced interval is cheap,  $x \in \Gamma$*
2. *Induced interval on line  $j$  does not contain any projection  $\lambda(c, l)$  of any  $c \in C$ .*

3. The same center from  $C$  serves all points in the induced interval,  $\exists_{c \in C} \forall_{y \in \xi(x)} y \in C'(c)$

4. Induced interval is either contained in  $F_\ell(C, X)$ , or  $F_\ell(C, X) \cap \xi(x) = \emptyset$

Then,  $\gamma(x) = \pi(x)$ . Otherwise,  $\gamma(x) = x$ .

**Lemma 3.4.**

$$|\text{cost}_\ell(D, C) - \text{cost}_\ell(Z, C)| \leq \mathcal{O}(\epsilon) \cdot (s + \sum_{j \in \mathcal{L}} t_j)$$

The strategy to show this lemma is to analyze the cost of intervals, which are breaking at least one of the conditions from [Definition 3.3](#). Before proving the bound, we require one technical lemma we adapt directly from [\[5\]](#) and leave unproven.

**Lemma 3.5** (Lemma 3.2 of [\[5\]](#)). *Let  $X = \{x_1, \dots, x_n\}$  and  $Y = \{y_1, \dots, y_n\}$  be two sequences of numbers. Then, for all  $S \subseteq [n]$ :*

$$|\text{top}_\ell(X) - \text{top}_\ell(Y)| \leq \ell \cdot \max_{i \in S} |x_i - y_i| + \sum_{i \in [n] \setminus S} |x_i - y_i|$$

where  $\text{top}_\ell(Z)$  is the sum of the  $\ell$  largest elements in  $Z$ .

Using [Lemma 3.5](#), we split the upper bound for  $|\text{cost}_\ell(D, C) - \text{cost}_\ell(Z, C)|$  into the sum of two terms (cost of *cheap* and *short* intervals) we analyze separately:

$$\begin{aligned} & |\text{cost}_\ell(D, C) - \text{cost}_\ell(Z, C)| \\ & \leq \ell \cdot \max_{x \in \Psi} |\text{dist}(\pi(x), C) - \text{dist}(\gamma(x), C)| + \sum_{x \in \Gamma} |\text{dist}(\pi(x), C) - \text{dist}(\gamma(x), C)| \end{aligned}$$

**Lemma 3.6** (Intervals breaking 1. condition).  $\ell \cdot \max_{x \in \Psi} |\text{dist}(\pi(x), C) - \text{dist}(\gamma(x), C)| \leq \mathcal{O}(\epsilon) \cdot s$

*Proof.* Observe,  $\forall x \in \Psi$ ,  $|\text{dist}(\pi(x), x)| \leq \text{len}(\xi(x)) = \mathcal{O}(\frac{\epsilon}{\ell} \cdot s)$ . Due to the triangle inequality:

$$\ell \cdot \max_{x \in \Psi} |\text{dist}(\pi(x), C) - \text{dist}(\gamma(x), C)| \leq \ell \cdot \mathcal{O}(\frac{\epsilon}{\ell} \cdot s) = \mathcal{O}(\epsilon) \cdot s$$

□

**Lemma 3.7** (Intervals breaking 2., 3. or 4. condition).  $\sum_{x \in \Gamma} |\text{dist}(\pi(x), C) - \text{dist}(\gamma(x), C)| \leq \mathcal{O}(\epsilon) \cdot \sum_{j \in \mathcal{L}} t_j$

*Proof.* This proof is a result of an observation that there are at most  $\mathcal{O}(k)$  sub-intervals  $\mathcal{Y}'_j \subseteq \mathcal{Y}_j$  induced by clients  $x \in \Gamma$  on each line  $j \in \mathcal{L}$ , such that  $\gamma(x) = x$ . To show this, we use the following propositions.

**Proposition 3.8** (Intervals breaking 2. condition). *There are at most  $k$  sub-intervals  $\hat{\mathcal{Y}}_j \subseteq \mathcal{Y}_j$  such that  $\lambda(c, l) \in Y$  for some  $c \in C$  and  $Y \in \hat{\mathcal{Y}}_j$*

*Proof.* Note that  $|C| = k$ . The number of projections into line  $j$  is also  $k$ . Hence,  $|\hat{\mathcal{Y}}_j| \leq k$ .  $\square$

**Proposition 3.9** (Intervals breaking 3. condition). *There are at most  $2k$  sub-interval  $\bar{\mathcal{Y}}_j \subseteq \mathcal{Y}_j$  belonging to two different clusters.*

*Proof.* For this proof, we first state the following observation.

**Observation 3.10.** *Spheres of any radius  $r > 0$  in  $\mathbf{L}_p$  for  $n \geq 1$  are convex figures.*

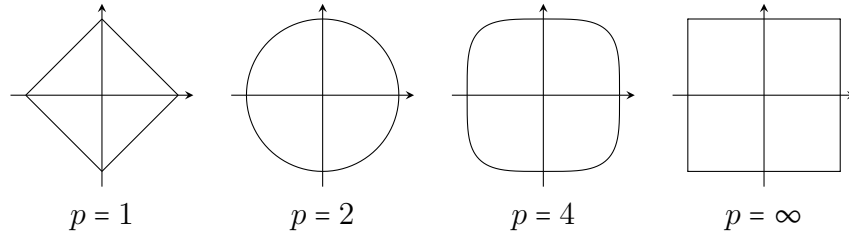


Figure 5: Unit spheres for in  $\mathbf{L}_p$  for different parameters  $p$ .

Using the observation above, we perform a procedure for determining the area of clusters  $C'(c)$  for every  $c \in C$ . Firstly, we create an  $r = 0$  sphere for each center. Then, we grow all spheres at the same time. We mark areas when spheres start intersecting each other in an uncovered area. We obtain *continuous* clustering in space. Each sphere (and therefore, cluster) intersects each  $j \in \mathcal{L}$  at most two times. The number of spheres is  $k$ , thus  $|\bar{\mathcal{Y}}_j| \leq 2k$ .  $\square$

**Proposition 3.11** (Intervals breaking 4. condition). *There are at most  $2k$  sub-intervals  $\tilde{\mathcal{Y}}_j \subseteq \mathcal{Y}_j$  such that  $Y \in \tilde{\mathcal{Y}}_j$  is not contained fully in  $F_\ell(C, X)$ , or  $F_\ell(C, X) \cap Y \neq \emptyset$*

*Proof.*  $\tilde{\mathcal{Y}}_j$  are sub-intervals located in intersection of  $F_\ell(C, X)$  and  $X \setminus F_\ell(C, X)$ . Similar to the previous proposition, we determine the set  $X \setminus F_\ell(C, X)$ . We continuously grow spheres from each of the centers in  $C$ . We stop when we have only  $\ell$  points outside the covered area. Following [Observation 3.10](#), we observe that in the worst case, all spheres intersect with line  $j$ , and none of the spheres intersect on line  $j$ . Then, sub-intervals on boundaries on the given area belong to  $\tilde{Y}$  and  $|\tilde{Y}_j| \leq 2k$ . If spheres are not intersecting  $j$  or intersecting on  $j$ , the number of sub-intervals in  $\tilde{Y}_j$  decreases. Hence,  $|\tilde{Y}_j| \leq 2k$ .  $\square$

Due to the union bound,  $\mathcal{Y}_j = \hat{\mathcal{Y}}_j \cup \bar{\mathcal{Y}}_j \cup \tilde{\mathcal{Y}}_j \implies |\mathcal{Y}_j| \leq \mathcal{O}(k)$ . Hence,

$$\begin{aligned} \sum_{x \in \Gamma} |\text{dist}(\pi(x), C) - \text{dist}(\gamma(x), C)| &= \sum_{x \in \Gamma: \gamma(x)=x} |\text{dist}(\pi(x), C) - \text{dist}(\gamma(x), C)| \\ &\leq \sum_{x \in \Gamma: \gamma(x)=x} \text{dist}(\pi(x), x) \leq \sum_{x \in \Gamma: \gamma(x)=x} \delta(\xi(x)) \\ &\leq \sum_{j \in \mathcal{L}} \mathcal{O}(k) \cdot \frac{\epsilon}{k} \cdot t_j = \mathcal{O}(\epsilon) \cdot \sum_{j \in \mathcal{L}} t_j \end{aligned}$$

This concludes the proof of [Lemma 3.7](#).  $\square$

*Proof of Lemma 3.4.* The proof is immediate from Lemma 3.6 and Lemma 3.7.  $\square$

**Lemma 3.12.**

$$|\text{cost}_\ell(Z, C) - \text{cost}_\ell(X, C)| \leq \mathcal{O}(\epsilon) \cdot d \cdot \sum_{j \in \mathcal{L}} t_j$$

Denote  $\mathcal{Y}' \subseteq \cup_{j \in \mathcal{L}} \mathcal{Y}_j$  be the series of intervals such that conditions from Definition 3.3 holds, hence  $\gamma(x) = \pi(x)$  for  $x \in X$  such that  $\xi(x) = Y$  for  $Y \in \mathcal{Y}'$ . This is sufficient to analyze this replacement error (incurred during moving point to its coreset point) to bound the difference above. Clients not belonging to sub-intervals in  $\mathcal{Y}'$  are not transformed  $\gamma$  operation mapping elements from  $X$  into the set  $Z$ . Thereby, they are not contributing to this difference.

For the sanity of deductions below, we need to assume that cumulative error from the interval is either contributing entirely or not contributing to the cost. In other words, we assume that intervals, which include points displaced during the projection  $\gamma$ , are entirely in  $F_\ell(C, Z)$  or  $F_\ell(C, X)$  or not intersecting at all. Then, we do not need to analyze the error contribution of intervals partially contributing to the cost. This holds due to Lemma 3.13.

Let  $F_\ell(C, Z) \subseteq Z$  be the set of the  $\ell$  furthest points of  $Z$  from  $C$ . Consider  $\mu(Y)$  as a set of all coreset points (concentrated in one position) displaced from their original positions.

**Lemma 3.13.**  $\mu(Y) \subseteq F_\ell(C, Z)$  or  $\mu(Y) \cap F_\ell(C, Z) = \emptyset$  for  $Y \in \mathcal{Y}'$

*Proof.* Due to the definition of  $\mathcal{Y}'$ , all points in  $Y \in \mathcal{Y}'$  are served by the same center  $c \in C$  (belongs to  $C'(c)$ ). Assume  $Y \in l$  for some  $j \in \mathcal{L}$ . Also, by definition,  $\lambda(c, l) \notin Y$ . Let  $a$  and  $b$  be endpoints of  $Y$  and assume, w.l.o.g., that  $\lambda(c, l)$  is closer to  $b$  than  $a$ . Using Lemma 3.14 we conclude that  $Y \cap F_\ell(C, X) = \emptyset$  implies  $\mu(Y) \cap F_\ell(C, Z) = \emptyset$  and  $Y \subseteq F_\ell(C, X)$  implies  $\mu(Y) \subseteq F_\ell(C, Z)$ .  $\square$

**Lemma 3.14.** Let  $a$  and  $b$  be endpoints of the sub-interval  $Y$  on line  $j$ . let  $c$  be a point (possibly outside of the line  $j$ ) with  $\lambda(c, l) \notin Y$  and  $\text{dist}(c, b) \leq \text{dist}(c, a)$ . Let  $e$  be between  $a$  and  $b$ . Then  $\text{dist}(c, b) \leq \text{dist}(c, e) \leq \text{dist}(c, a)$ .

*Proof.* Assume w.l.o.g, that  $\text{dist}(c, b) < \text{dist}(c, a)$ . Equality possibly holds only if the spheres are not *strictly convex* (i.e., they are visual *squares* in  $\mathbf{L}_1$  and  $\mathbf{L}_\infty$ ), and the sub-interval  $Y$  is fully included in the edge of the sphere. Those trivial cases do not require special attention.

Following the Observation 3.10, we observe that the sphere of size  $\text{dist}(c, \lambda(c, l))$  has only one common point with  $j$  (i.e.,  $\lambda(c, l)$ ). While growing the sphere, we will constantly have two intersection points on the line  $j$  – one on each side from  $\lambda(c, l)$ . Therefore, firstly, our growing sphere will cross  $b$ , then  $e$ , and ultimately  $c$ . Figure 6 depicts this procedure. Hence,  $\text{dist}(c, b) < \text{dist}(c, e) < \text{dist}(c, a)$ .  $\square$

*Proof of Lemma 3.12.* Due to Lemma 3.13, elements from  $Z \setminus F_\ell(Z, C)$  do not contribute to either  $\text{cost}_\ell(Z, C)$  or  $\text{cost}_\ell(X, C)$  (before displacing). Hence, this is enough to analyze the error in the intervals  $\mathcal{Y}^*$  induced by the subset of points in  $F_\ell(C, X)$ .

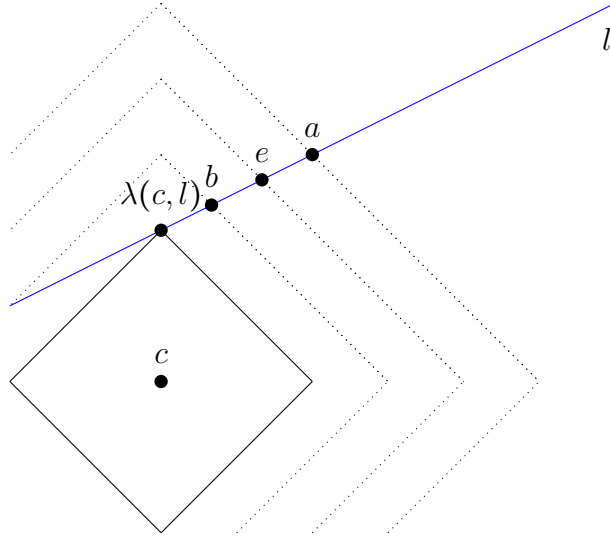


Figure 6: Growing spheres in  $\mathbf{L}_1$  (or  $\mathbf{L}_\infty$ ) crossing points in  $b$ ,  $e$  and  $a$ .  $c$  is lays in an origin.

By [Lemma 3.15](#) inspired by [58] the total replacement error for  $\bar{\mathcal{Y}}_{l,c} \subseteq \mathcal{Y}^*$  defined as subset of  $\mathcal{Y}^* \cap \mathcal{Y}_j$  served by  $c$  is at most  $\mathcal{O}(\frac{\epsilon}{k} \cdot t_j \cdot d)$ . Obviously,

$$\bigcup_{c \in C, j \in \mathcal{L}} \bar{\mathcal{Y}}_{l,c} = \mathcal{Y}^*$$

$\bigcup_{c \in C, j \in \mathcal{L}} \bar{\mathcal{Y}}_{l,c} = \mathcal{Y}^*$  Since  $|C| = k$ :

$$\begin{aligned} |\text{cost}_\ell(Z, C) - \text{cost}_\ell(X, C)| &\leq \sum_{j \in \mathcal{L}, c \in C} \mathcal{O}\left(\frac{\epsilon}{k} \cdot t_j \cdot d\right) \\ &= \mathcal{O}(\epsilon) \cdot d \cdot \sum_{j \in \mathcal{L}} t_j \end{aligned}$$

That concludes the proof of [Lemma 3.12](#). □

**Lemma 3.15.** *Replacement error of  $\bar{\mathcal{Y}}_{l,c}$  is at most  $\mathcal{O}(\frac{\epsilon}{k} \cdot t_j \cdot d)$ .*

To prove this lemma, we need one observation noticeable while growing coordinates on line  $j$  or by induction with the number of dimensions.

**Observation 3.16.** *In  $\mathbb{R}^d$ , line  $j$  is crossing at most  $d + 1$  orthants.*

*Proof of Lemma 3.15.* Using [Observation 3.16](#), we notice that there are at most  $d + 1$  intervals in  $\tilde{\mathcal{Y}}_{l,c} \subseteq \bar{\mathcal{Y}}_{l,c}$  that are contained in more than one orthants with the origin of the Euclidean space located in  $c$ . Their total replacement error is upper bounded by  $(d + 1) \cdot \mathcal{O}(\frac{\epsilon}{k} \cdot t_j) = \mathcal{O}(\frac{\epsilon}{k} \cdot t_j \cdot d)$

Remaining intervals are bucked into at most  $d + 2$  subsets  $\mathcal{Y}_{l,c,i,s}$  corresponding to a different  $i$  orthant. Additionally, we introduce  $s \in \{0, 1\}$  referring to the side from the  $\lambda(c, l)$  (for at most one  $i$ , sets for both values of  $s$  are non-empty).

$$\bigcup_{i \in [d+1], s \in \{0,1\}} \mathcal{Y}_{l,c,i,s} = \bar{\mathcal{Y}}_{l,c} \setminus \tilde{\mathcal{Y}}_{l,c} = \dot{\mathcal{Y}}_{l,c}$$

Using [Lemma 3.17](#), we observe that the total replacement error in  $\dot{\mathcal{Y}}_{l,c}$  is at most  $(d+2) \cdot \mathcal{O}(\frac{\epsilon}{k} \cdot t_j) \in \mathcal{O}(\frac{\epsilon}{k} \cdot t_j \cdot d)$ .

Since  $\dot{\mathcal{Y}}_{l,c} \cup \tilde{\mathcal{Y}}_{l,c} = \bar{\mathcal{Y}}_{l,c}$ , the total replacement error for  $\bar{\mathcal{Y}}_{l,c}$  is also  $\mathcal{O}(\frac{\epsilon}{k} \cdot t_j \cdot d)$ .  $\square$

**Lemma 3.17.** *Replacement error of  $\mathcal{Y}_{l,c,i,s}$  is at most  $\mathcal{O}(\frac{\epsilon}{k} \cdot t_j \cdot d)$ .*

Before proving this lemma, we need another technical lemma that we will consecutively use toward the rest of the subsection. We will refer to  $\mathcal{Y}_{l,c,i,s}$  as  $T$  to simplify deductions.

**Lemma 3.18.** *With any  $\mathbf{L}_p$  norm in the  $d$ -dimensional Euclidean space, for any set of points  $G \in \mathbb{R}^d$  located on the same line in the same orthant.*

$$\sum_{g \in G} \text{dist}(\mathbf{0}, g) \geq |G| \cdot \text{dist}\left(\mathbf{0}, \frac{\sum_{g \in G} g}{|G|}\right)$$

*Proof.* W.l.o.g, we assume that all points lay in the first orthant. Due to the norm subadditivity,  $\text{dist}(\mathbf{0}, a) + \text{dist}(\mathbf{0}, b) \leq \text{dist}(\mathbf{0}, a + b)$ . By induction,  $\sum_{g \in G} \text{dist}(\mathbf{0}, g) \geq \text{dist}(\mathbf{0}, \sum_{g \in G} g)$ . With *absolute homogeneity*, we obtain our desired inequality.  $\square$

Now, we are ready to prove [Lemma 3.17](#).

*Proof of Lemma 3.17.* Assume w.l.o.g that  $\delta(Y) < \frac{\epsilon}{k} \cdot t_j$  for all  $Y \in T$ . We can do this by multiplying  $\epsilon$  with some constant. This will not impact our outcome since  $\mathcal{O}(\epsilon) \cdot \mathcal{O}(1) \in \mathcal{O}(\epsilon)$ .

The replacement error is an error we occur by replacing points with their coreset equivalent. The *replacement error* for the set  $T$  of sub-intervals is defined as:

$$\text{err}(T) := \sum_{Y \in T} \left( \sum_{x \in X \cap Y} \text{dist}(c, x) \right) - |X \cap Y| \cdot \text{dist}(c, \mu(Y))$$

For each sub-interval  $Y \in T$ , we define the *partial replacement error* as:

$$\text{err}'(Y) := \left( \sum_{x \in X \cap Y} \text{dist}(c, x) \right) - |X \cap Y| \cdot \text{dist}(c, \mu(Y))$$

Following [Lemma 3.18](#),  $\text{err}'(Y) \geq 0$ , hence  $\text{err}(T) \geq 0$ . Thus, it remains to prove that  $\text{err}(T) \leq \mathcal{O}(\frac{\epsilon}{k} \cdot t_j)$ . Instead of directly bounding the error in  $T$ , we create a modified instance  $\tilde{T}$  from  $T$  that is having following properties:

- $\forall Y \in \tilde{T} \delta(Y) = \frac{\epsilon}{k} \cdot t_j$
- $\text{err}(\tilde{T}) \geq \text{err}(T)$

Then, this is enough to bound the error in  $\tilde{T}$ .

We first remove all sub-intervals having exactly one client from  $X$  to obtain this instance. For those sub-intervals,  $\text{err}' = 0$ , so we can safely do it. Next, to each  $Y \in T$ , we add points  $a$  and  $b$ , so that the coreset point  $\mu(Y)$  is equally close to both points.

We can do this due to the assumption that  $Y$  is not a singleton. Then, we multiply those points (obtaining set  $A$  and  $B$  for each  $Y$ ) until  $\forall Y \in T \delta(Y) \geq \frac{\epsilon}{k} \cdot t_j$ . After meeting this threshold, we move those points closer together until  $\forall Y \in T \delta(Y) = \frac{\epsilon}{k} \cdot t_j$ . During this operation, we only increase the replacement error. We assign the set of modified (and reduced) sub-intervals to  $\tilde{T}$ . We will denote the cumulative error for each  $Y \in \tilde{T}$ ,  $\delta(Y) = \Delta$ .

We split sub-intervals from  $\tilde{T}$  into  $M_i$  and  $N_i$  such that  $M_i \cup N_i = Y_i$  and  $M_i$  is located on the one side of  $\mu(Y_i)$  – closer to  $c$ .  $N_i$  stays on the opposite side. They are sorted by the distance from  $c$  as presented in [Figure 7](#).

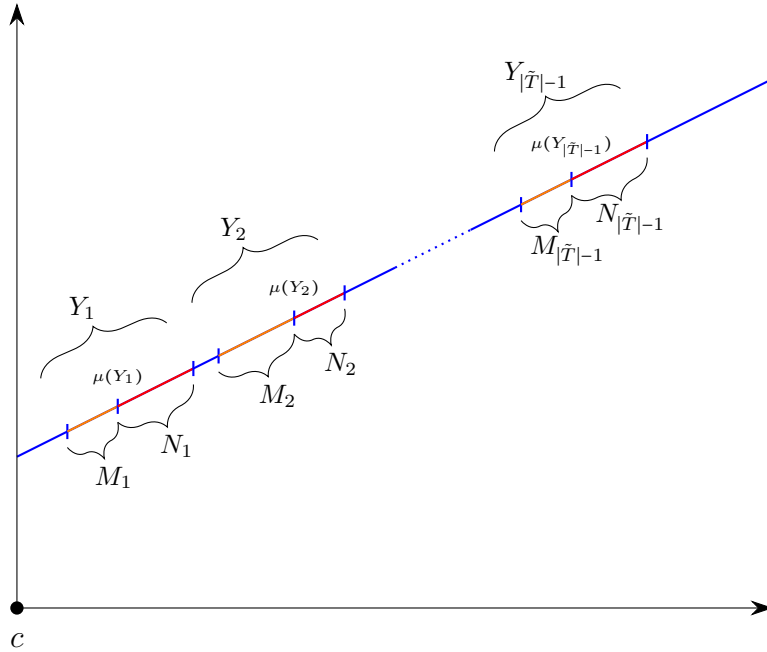


Figure 7: Splitting the sub-intervals' set  $\tilde{T}$

Let  $\mu'(M_i)$  or  $\mu'(N_i)$  denote set of points displaced to the coreset point  $\mu(Y_i)$  during  $\gamma$  operation. Let  $\chi(M_i)$ ,  $\chi(N_i)$ ,  $\chi(\mu'(M_i))$ , and  $\chi(\mu'(N_i))$  be the cost incurred by segments.

$$err(\tilde{T}) = \sum_{i=1}^{|\tilde{T}|} \chi(M_i) + \chi(N_i) - \chi(\mu'(M_i)) - \chi(\mu'(N_i))$$

Because of [Lemma 3.19](#) stated below,

$$\chi(N_i) - \chi(\mu'(N_i)) + \chi(M_{i+1}) - \chi(\mu'(M_{i+1})) \leq 0$$

Hence,

$$err(\tilde{T}) \leq \chi(M_1) - \chi(\mu'(M_1)) + \chi(N_{|\tilde{T}|}) - \chi(\mu'(N_{|\tilde{T}|}))$$

Thus, because  $err(T) \geq 0$ ,

$$err(T) \in \mathcal{O}\left(\frac{\epsilon}{k} \cdot t_j\right)$$

That concludes the [Lemma 3.17](#) □

**Lemma 3.19.**

$$\chi(\mu'(N_i)) - \chi(N_i) \geq \chi(M_{i+1}) - \chi(\mu'(M_{i+1}))$$

*Proof.* Since displacement of  $M_i$  and  $N_{i+1}$  is contributing to the same cumulative error (namely,  $\frac{\Delta}{2}$ ) and all point within  $M_i$  and  $N_{i+1}$  are collocated, we see that for some positive  $\alpha$ :

$$\alpha \cdot dist(\mu(N_i), N_i) = dist(\mu(M_{i+1}), M_{i+1})$$

And:

$$|N_i| = |M_{i+1}| \cdot \alpha$$

Also, they are ordered on a line  $j$  (by distance from  $c$ )

$$dist(c, \mu(N_i)) \leq dist(c, N_i) \leq dist(c, M_{i+1}) \leq dist(c, \mu(M_{i+1}))$$

Hence, this is enough to prove that:

$$\alpha \cdot dist(c, \mu(N_i)) - \alpha \cdot dist(c, N_i) \geq dist(c, M_{i+1}) - dist(c, \mu(M_{i+1}))$$

[Figure 8](#) depicts this construction.

We can assume that  $N_i = M_{i+1}$  (are collocated). If this is not true, it does not require additional effort to prove because we can displace the interval between  $M_{i+1}$  and  $\mu(M_{i+1})$  obtaining even a stronger inequality by [Proposition 3.20](#). Therefore, this is enough to prove:

$$\alpha \cdot dist(c, \mu(N_i)) + dist(c, \mu(M_{i+1})) \geq (1 + \alpha) \cdot dist(c, N_i).$$

Observe that:

$$N_i = M_{i+1} = \frac{\sum_{g \in \mu(N_i) \cup \mu(M_{i+1})} g}{|\mu(N_i) \cup \mu(M_{i+1})|}$$

Thus, due to [Lemma 3.18](#),

$$|N_i| \cdot dist(c, \mu(N_i)) + |M_{i+1}| \cdot dist(c, \mu(M_{i+1})) \geq |N_i \cup M_{i+1}| \cdot dist(c, N_i)$$

After dividing each sides of the inequality by  $|M_{i+1}|$  and using the fact that  $|N_i| = |M_{i+1}| \cdot \alpha$ , the expected inequality is proved. □

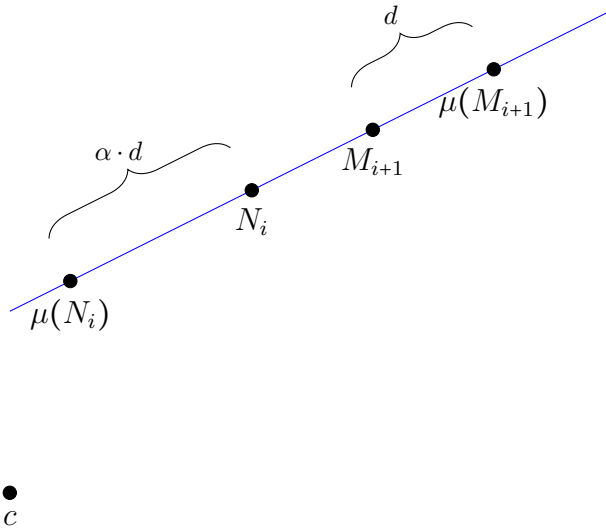


Figure 8: Points  $\mu(N_i)$ ,  $N_i$ ,  $M_{i+1}$ ,  $\mu(M_{i+1})$  on a line with within the same orthant with the same weighted distance between  $\mu(N_i)$ ,  $N_i$ , and  $M_{i+1}$ ,  $\mu(M_{i+1})$ .

**Proposition 3.20.** *If  $a, b, a', b'$  and points in  $\mathbb{R}^d$  staying in one orthant on line  $j$  such that  $\text{dist}(\mathbf{0}, a) \leq \text{dist}(\mathbf{0}, b)$ ,  $\text{dist}(\mathbf{0}, a') \leq \text{dist}(\mathbf{0}, b')$ ,  $\text{dist}(\mathbf{0}, a) \leq \text{dist}(\mathbf{0}, a')$  and  $\text{dist}(a, b) = \text{dist}(a', b')$ , then  $\text{dist}(\mathbf{0}, b') - \text{dist}(\mathbf{0}, a') \geq \text{dist}(\mathbf{0}, b) - \text{dist}(\mathbf{0}, a)$ .*

*Proof.* Consider this inequality as a displacement of the interval  $\|ab\|$  into a new position  $\|a'b'\|$ . Assume that  $\text{dist}(a, b) = \text{dist}(a', b') = t_1$ ,  $\text{dist}(a, a') = \text{dist}(b, b') = t_2$  as in Figure 9.

Assume w.l.o.g. that  $t_1, t_2 \in \mathbb{Q}^+$ . Then we can discretize  $\|ab\|$  and  $\|a'b'\|$  into an integral number of subintervals of length  $t_3$  such that each needs to be displaced an integral number of times by distance  $t_3$  to match its equivalent position in a second sub-interval. Each of those small displacements is a special case of Lemma 3.18, sufficient to conclude the proof. In this case, we can use Lemma 3.18 in the limited version without relying on the currently being proved proposition because the interval is displaced directly next to the previous position.  $\square$

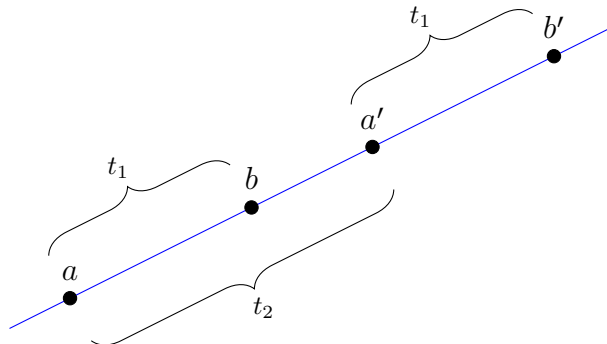


Figure 9: Points  $a, b, a', b'$  on a line with within the same orthant with distances between them.

### 3.4 Non-simultaneous coresets for $\ell$ -centrum

The step forward from [Theorem 3.2](#) is using those coresets to construct  $\epsilon$ -coresets for  $\ell$ -CENTRUM problem. Again, we use a procedure inspired by [\[58\]](#) and [\[5\]](#) that gives us an expected result without severe modifications.

**Theorem 3.21.** *For any values  $k, \ell \in \mathbb{Z}^+$ , data set  $X \subseteq \mathbb{R}^d$ , and any  $\epsilon \in (0, 1)$ , there is a weighted set  $D$  of size  $\mathcal{O}(\frac{k^2 \cdot d}{\epsilon^{d+1}})$  such that:*

$$\text{cost}_\ell(D, C) \in (1 \pm \mathcal{O}(\epsilon)) \cdot \text{cost}_\ell(X, C)$$

for all sets  $C \subseteq \mathbb{R}^d$  of size  $k$  and  $D$  is obtained in polynomial time.

First, we need to define the projection procedure that reduces any instance in  $\mathbb{R}^d$  to the lines-based case. Let  $\tilde{C}$  denote the projection centers we will define conveniently later.

#### Projection scheme

1. For each center  $c$  from  $\tilde{C}$ , define  $N(c)$  that is  $\epsilon$ -net (using the definition from [\[59\]](#)) unit sphere ( $r = 1$ ) centered in  $c$ .
2. Then, for each  $v \in N(c)$  for  $c \in \tilde{C}$ , define  $l(v, c)$  to be a line passing through those points. Denote the set of those lines as  $\mathcal{L}$ .
3. Define  $\tilde{X}$  as a projected set from  $X$  to the closest line projected from the closest center  $c \in \tilde{C}$ . Precisely,  $\tilde{X} = \{\tilde{\lambda}(x) \mid x \in X\}$ , where  $\tilde{\lambda}(x)$  for  $x \in X$  denotes the projection (and  $\tilde{\lambda}(x)^{-1}$  the opposite operation) defined as  $\tilde{\lambda}(x) = \lambda(x, l(v, c))$ , where  $v = \arg \min_{v' \in N(c)} \text{dist}(l(v', c), x)$  and  $c = \arg \min_{c \in \tilde{C}} \text{dist}(x, c)$ .

Since  $N(c)$ 's are  $\epsilon$ -net spheres in  $\mathbb{R}^d$ ,  $|\mathcal{L}| \leq \mathcal{O}(\frac{1}{\epsilon})^d \cdot |\tilde{C}|$  due to [\[60\]](#) in  $\mathbf{L}_p$  for all  $p$ . [Figure 10](#) depicts the projection with  $\epsilon$ -nets.

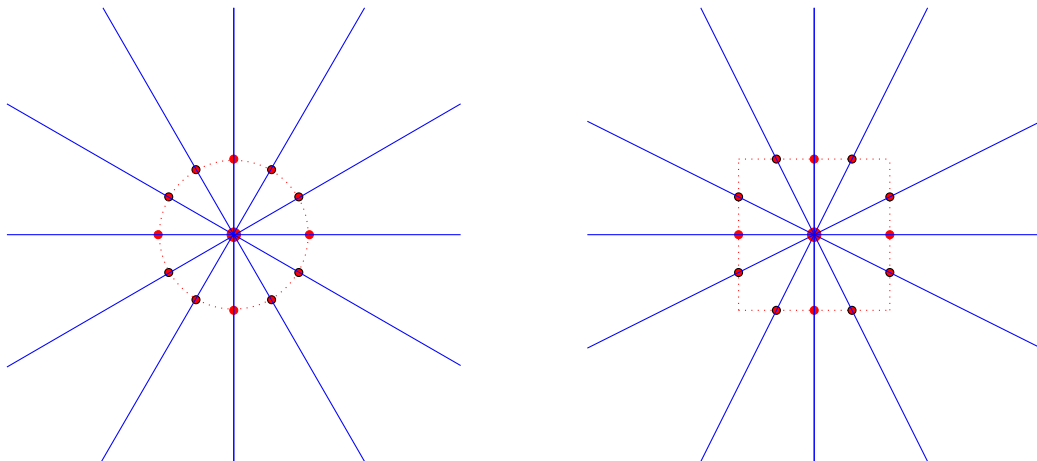


Figure 10:  $\epsilon$ -nets for unit spheres in  $\mathbb{R}^2$  for  $\mathbf{L}_2$  and  $\mathbf{L}_\infty$  with lines projections.

**Lemma 3.22.** *For all  $C \in \mathbb{R}^d$  and  $\ell \leq |X|$ ,  $|\text{cost}_\ell(\tilde{X}, C) - \text{cost}_\ell(X, C)| \leq \mathcal{O}(\epsilon) \cdot \text{cost}_\ell(X, \tilde{C})$*

*Proof.* Observe that by triangle inequality (given by the metricity)  $\text{dist}(x, C) - \text{dist}(\tilde{\lambda}(x), C) \leq \text{dist}(x, \tilde{\lambda}(x))$  for  $x \in X$ . Also, due to the definition of the projection and  $\epsilon$ -nets.  $\text{dist}(x, \tilde{\lambda}(x)) \leq \mathcal{O}(\epsilon) \cdot \text{dist}(x, \tilde{C})$ . Let  $F_\ell(C, \tilde{X})$  be the  $\ell$  furthest points from  $C$  in  $\tilde{X}$ . Similarly, let  $F_\ell(C, X)$  be the  $\ell$  furthest points from  $C$  in  $X$ .

$$\begin{aligned} \text{cost}_\ell(\tilde{X}, C) &= \sum_{x \in F_\ell(C, \tilde{X})} \text{dist}(x, C) \\ &\leq \sum_{x \in F_\ell(C, \tilde{X})} \text{dist}(\tilde{\lambda}^{-1}(x), C) + \mathcal{O}(\epsilon) \sum_{x \in F_\ell(C, \tilde{X})} \text{dist}(\tilde{\lambda}^{-1}(x), \tilde{C}) \\ &\leq \text{cost}_\ell(X, C) + \mathcal{O}(\epsilon) \cdot \text{cost}_\ell(X, \tilde{C}) \end{aligned}$$

Similarly,

$$\begin{aligned} \text{cost}_\ell(X, C) &= \sum_{x \in F_\ell(C, X)} \text{dist}(x, C) \\ &\leq \sum_{x \in F_\ell(C, X)} \text{dist}(\tilde{\lambda}(x), C) + \mathcal{O}(\epsilon) \sum_{x \in F_\ell(C, X)} \text{dist}(x, \tilde{C}) \\ &\leq \text{cost}_\ell(\tilde{X}, C) + \mathcal{O}(\epsilon) \cdot \text{cost}_\ell(X, \tilde{C}) \end{aligned}$$

□

Now, we are ready to prove [Theorem 3.21](#)

*Proof of Theorem 3.21.* First, we use any  $\mathcal{O}(1)$  polynomial time algorithm for  $\ell$ -CENTRUM to select centers  $\tilde{C}$ . For example, we can adapt results of [26] that is giving us  $\mathcal{O}(1) \cdot \text{OPT}$  (precisely,  $3\text{OPT}$ ) solution, where  $\text{OPT}$  is the optimal solution for  $\ell$ -CENTRUM. Then, the error incurred by projection procedure from set  $X$  to  $\tilde{X}$  is  $\mathcal{O}(\epsilon) \cdot \text{OPT}$ .

**Definition 3.23** (Sub-intervals' splitting). *For each  $j \in \mathcal{L}$  we define sub-interval as follows. Let  $S_j = \tilde{X} \cap j$ . Then,  $S_1$  is the subset of  $\ell$  furthest points from  $\tilde{C}$ ,  $S_{j,1} = S_1 \cap j$  and  $S_{j,2}$  be the compliment of this set,  $S_{j,2} = S_j \setminus S_{j,1}$ . Let  $\text{APX} := \text{cost}_\ell(\tilde{X}, C)$  and let  $\text{APX}_j$  be the contribution to  $\text{APX}$  over the line  $j$ . We greedily break  $S_{1,1}$  into maximal sub-intervals with error  $\delta(Y) \in \mathcal{O}(\frac{\epsilon \cdot \text{APX}_j}{k \cdot d})$ . Similarly, we break  $S_{1,2}$  into sub-intervals of length  $\mathcal{O}(\frac{\epsilon \cdot \text{APX}}{\ell \cdot d})$ .*

The number of sub-intervals is at most  $\mathcal{O}(\frac{k \cdot d}{\epsilon})$  for each line  $j \in \mathcal{L}$  due to following analysis. Eventually, we assign  $t_j = \frac{\text{APX}_j}{d}$  and  $s = \frac{\text{APX}}{d}$ , use subsets splitting obtained by the procedure described above, and apply [Theorem 3.2](#) to obtain the set  $D$ .

$$\begin{aligned} |\text{cost}_\ell(D, C) - \text{cost}_\ell(X, C)| &\leq \mathcal{O}(\epsilon) \cdot \text{OPT} + |\text{cost}_\ell(D, C) - \text{cost}_\ell(\tilde{X}, C)| \\ &= \mathcal{O}(\epsilon) \cdot \text{OPT} + \mathcal{O}(\epsilon \cdot d) \cdot \left( \frac{\text{APX}}{d} + \sum_{j \in \mathcal{L}} \frac{\text{APX}_j}{d} \right) \\ &= \mathcal{O}(\epsilon) \cdot \text{OPT} + \mathcal{O}(\epsilon \cdot d) \cdot \frac{\text{OPT}}{d} = \mathcal{O}(\epsilon) \cdot \text{OPT}. \end{aligned}$$

Also,

$$|D| \leq |\mathcal{L}| \cdot \mathcal{O}\left(\frac{k \cdot d}{\epsilon}\right) \leq \mathcal{O}\left(\frac{k \cdot d}{\epsilon^{d+1}}\right) \cdot |C| = \mathcal{O}\left(\frac{k^2 \cdot d}{\epsilon^{d+1}}\right)$$

That concludes the proof of [Theorem 3.21](#).  $\square$

### 3.5 Coreset for $k$ -Median

The obvious corollary from the previous results is at the same time generalization of [\[58\]](#) theorem regarding  $k$ -MEDIAN coreset.

Denote  $\text{cost} := \text{cost}_n$ , where  $n$  is the size of the clients set. Particularly, this is the cost of the  $k$ -MEDIAN objective. Then, let  $\ell = n$ .

**Corollary 3.24.** *For any value  $k \in \mathbb{Z}^+$ , data set  $X \subset \mathbb{R}^d$  of size  $|X| = n$ , and any  $\epsilon \in (0, 1)$ , there is a weighted set  $D$  of size  $\mathcal{O}\left(\frac{k^2 \cdot d}{\epsilon^{d+1}}\right)$  such that:*

$$\text{cost}(D, C) \in (1 \pm \mathcal{O}(\epsilon)) \cdot \text{cost}(X, C)$$

for all sets  $C \subset \mathbb{R}^d$  of size  $k$ . Furthermore,  $D$  is computed in polynomial time.

### 3.6 Simultaneous coreset for Ordered $k$ -Median

Consolidating previous results is enough to prove [Theorem 3.1](#) about simultaneous coreset for  $\ell$ -CENTRUM using bucketing with different values of  $\ell$ . Before describing the bucketing procedure, we need one lemma that allows for bounding the cost within different values of  $\ell$  that are close to each other within a logarithmic multiplicative factor.

**Lemma 3.25.** *For each  $\ell, q \leq n$ , such that  $\ell \leq q$  and  $q \leq (1 + \epsilon) \cdot \ell$ ,  $X, C \subset \mathbb{R}^d$*

$$\text{cost}_\ell(X, C) \leq \text{cost}_q(X, C) \leq (1 + \epsilon) \cdot \text{cost}_\ell(X, C)$$

*Proof.* Obviously,  $\text{cost}_\ell(X, C) \leq \text{cost}_q(X, C)$  since we are considering more clients. However, the number of clients is at most  $(1 + \epsilon)$  times greater, and the weights are decreasing write formulating the objective, so  $\text{cost}_q(X, C) \leq (1 + \epsilon) \cdot \text{cost}_\ell(X, C)$ .  $\square$

Denote  $W_\delta := \{(1 + \delta)^i, i \in \{1, \dots, \lfloor \log_{1+\delta} n \rfloor\}\} \cup \{n\}$

**Lemma 3.26.** *For  $X \in \mathbb{R}^d$  such that  $|X| = n$  and some  $t \in (0, 1)$ , set  $D$  that is a non-simultaneous  $\epsilon$ -coreset for  $\ell$ -CENTRUM for all  $\ell \in W_{\frac{\epsilon}{1-t}}$ , i.e.,*

$$\forall_{C \in \mathbb{R}^d, |C|=k} \forall_{\ell \in W_{\frac{\epsilon}{1-t}}} \text{cost}_\ell(D, C) \in (1 \pm \mathcal{O}(\epsilon)) \cdot \text{cost}_\ell(X, C)$$

is also a simultaneous  $\epsilon$ -coreset for the ORDERED  $k$ -MEDIAN problem instances specified by  $v$  restricted by given  $t \in (0, 1)$ , i.e.,

$$\forall_{C \in \mathbb{R}^d, |C|=k} \forall_{v \in (t, 1)^n} \text{cost}_v(D, C) \in (1 \pm \mathcal{O}(\epsilon)) \cdot \text{cost}_v(X, C)$$

*Proof.* Denote  $\mathbf{t}$  a vector of size  $n$  containing only elements  $t$ . Denote  $v - t$  a vector  $v$  with all elements decreased by  $t$ , i.e.,  $\forall_{i \in \{1, \dots, n\}} (v - t)_i = v_i - t$ . Observe that  $\mathbf{t} + (v - t) = v$ , with the element-wise sum. For all centers,  $C$  of size  $k$  and vector  $v$  restricted by  $t$ . Here, we assume that  $D$  is not a weighted set, but points are collocated.

$$\begin{aligned} \text{cost}_v(D, C) &= \sum_{i=1}^n v_i \cdot \max_{x \in D}^i \text{dist}(x, C) \\ &\in (1 \pm \epsilon) \cdot \left( \sum_{i=1}^{n-1} (v - t)_i \cdot \max_{x \in D}^i \text{dist}(x, C) + \sum_{i=1}^n \mathbf{t}_i \cdot \max_{x \in D}^i \text{dist}(x, C) \right) \end{aligned}$$

Let:

$$\begin{aligned} A &= \sum_{i=1}^{n-1} (v - t)_i \cdot \max_{x \in D}^i \text{dist}(x, C) \\ B &= \sum_{i=1}^n \mathbf{t}_i \cdot \max_{x \in D}^i \text{dist}(x, C) \end{aligned}$$

Observe:

$$\sum_{i=1}^n \mathbf{t}_i \cdot \max_{x \in D}^i \text{dist}(x, C) = t \cdot \sum_{i=1}^n 1 \cdot \max_{x \in D}^i \text{dist}(x, C)$$

Using definition of  $W_{\frac{\epsilon}{1-t}}$  and [Lemma 3.25](#).

$$\begin{aligned} A &= \sum_{i=1}^{n-1} (v - t)_i \cdot \max_{x \in D}^i \text{dist}(x, C) \\ &= \sum_{\ell=1}^{n-1} ((v_\ell - t) - (v_{\ell+1} - t)) \cdot \sum_{i=1}^{\ell} \max_{x \in D}^i \text{dist}(x, C) \\ &= \sum_{\ell=1}^{n-1} (v_\ell - v_{\ell+1}) \cdot \sum_{i=1}^{\ell} \max_{x \in D}^i \text{dist}(x, C) \\ &\in \left( 1 \pm \mathcal{O}\left(\frac{\epsilon}{1-t}\right) \right) \cdot (1 \pm \mathcal{O}(\epsilon)) \cdot \sum_{i=1}^n (v - t)_i \cdot \max_{x \in X}^i \text{dist}(x, C) \\ &\in \left( 1 \pm \mathcal{O}\left(\frac{\epsilon}{1-t}\right) \right) \cdot \sum_{i=1}^n (v - t)_i \cdot \max_{x \in X}^i \text{dist}(x, C) \end{aligned}$$

Eventually, we need the following observation to conclude obtaining an expected bound.

**Observation 3.27.**  *$B$  is contributing to at least  $t$  part of the  $\text{cost}_v$ , so  $A$  is contributing to at most  $1 - t$  part of the sum*

The observation becomes clear after noticing that the vector  $v$  divides into  $A$  and  $B$ , and the distances considered in a sum are in decreasing order. [Figure 11](#) depicts this idea.

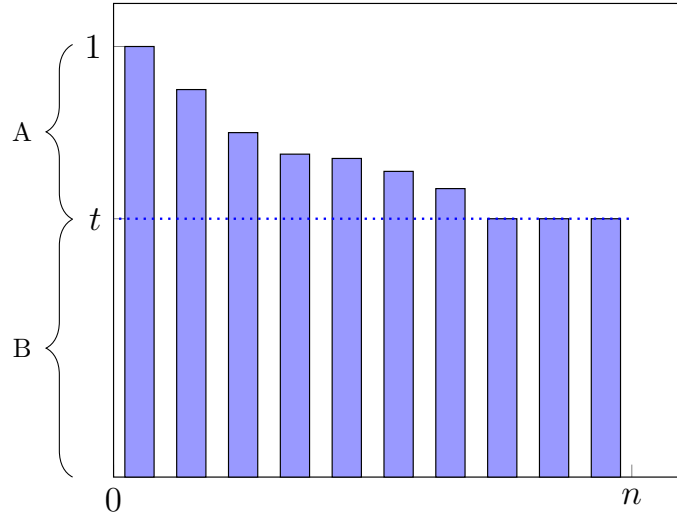


Figure 11: The vector  $v$  is representing decreasing weights that are split into  $A$  and  $B$  by line defined by line on  $t$ .

Taking all of those observations into consideration,

$$\begin{aligned}
\text{cost}_v(D, C) &= A + B \\
&\in (1 \pm \mathcal{O}(\frac{\epsilon}{1-t})) \cdot \sum_{i=1}^n (v-t)_i \cdot \max_{x \in X}^i \text{dist}(x, C) + (1 \pm \mathcal{O}(\epsilon)) \cdot \sum_{i=1}^n t_i \cdot \max_{x \in X}^i \text{dist}(x, C) \\
&\in ((1 \pm \mathcal{O}(\frac{\epsilon}{1-t})) \cdot (1-t) + (1 \pm \mathcal{O}(\epsilon))) \cdot \sum_{i=1}^n v_i \cdot \max_{x \in X}^i \text{dist}(x, C) \\
&= ((1 \pm \mathcal{O}(\frac{\epsilon}{1-t})) \cdot (1-t) + (1 \pm \mathcal{O}(\epsilon))) \cdot \text{cost}_v(X, C) \\
&\in (1-t \pm \mathcal{O}(\epsilon) + t \pm \mathcal{O}(\epsilon \cdot t)) \cdot \text{cost}_v(X, C) \\
&\in (1 \pm \mathcal{O}(\epsilon)) \cdot \text{cost}_v(X, C)
\end{aligned}$$

That concludes the proof of [Lemma 3.26](#)  $\square$

Hence, to eventually show [Theorem 3.1](#), we need to find coresets proper for all  $\ell \in W_{\frac{\epsilon}{1-t}}$ .

*Proof of [Theorem 3.1](#).* To find that, we use a very similar technique as in [Theorem 3.21](#). However, the projection centers needs to be an  $\mathcal{O}(1)$  approximation for all  $\ell \in W_{\frac{\epsilon}{1-t}}$ . Notice that  $\tilde{C}$  from the previous section does not need to be of size  $k$ . Greedily, let  $\tilde{C} = \cup_{\ell \in W_{\frac{\epsilon}{1-t}}} \tilde{C}_\ell$ . Let  $\tilde{C}_\ell$  be  $\mathcal{O}(1)$  approximation of  $\ell$ -CENTRUM for the given  $\ell$ . Observe,

$$\begin{aligned}
|\tilde{C}| &= k \cdot |W_{\frac{\epsilon}{1-t}}| \\
&\in \mathcal{O}(k \cdot (1 + \log_{1+\frac{\epsilon}{1-t}} n)) \\
&\in \mathcal{O}\left(k \cdot \left(1 + \frac{\log n}{\log(1 + \frac{\epsilon}{1-t})}\right)\right) \\
&\in \mathcal{O}\left(k \cdot \left(1 + \frac{(1-t) \cdot \log n}{\epsilon}\right)\right)
\end{aligned}$$

Then, we use a similar procedure and obtain  $D_\ell$  being a coresets for a specific  $\ell \in W_{\frac{\epsilon}{1-t}}$ . After replicating the same reasoning:

$$|D_\ell| = \mathcal{O}\left(\left(1 + \frac{(1-t) \cdot \log n}{\epsilon}\right) \cdot \frac{k \cdot d}{\epsilon^{d+1}}\right)$$

However, this is insufficient to obtain the final coresets as a union of all  $D_\ell$  for  $\ell \in W_{\frac{\epsilon}{1-t}}$ , because clients are served by more than one coresets point.

Notice that the coresets for each  $D_\ell$  was obtained using [Theorem 3.2](#) by defining sub-intervals on lines. For each  $D_\ell$  we use the same  $\tilde{C}$ , thus the same set of lines. Hence, we can freely replace those sub-intervals with their sub-intervals. For each  $\ell \in W_{\frac{\epsilon}{1-t}}$ , we obtain sub-intervals using a technique from [Theorem 3.21](#). Then, we use a sub-interval resulting from the intersections. Remarkably, every endpoint of each sub-interval divides other sub-intervals into two. The following inequality is a counting argument on the number of endpoints.

$$\begin{aligned} n &\leq |D_\ell| \cdot |W_{\frac{\epsilon}{1-t}}| \\ &\in \mathcal{O}\left(\left(1 + \frac{(1-t) \cdot \log n}{\epsilon}\right) \cdot \frac{k \cdot d \cdot \log n}{\epsilon^{d+1}}\right) \cdot \mathcal{O}\left(1 + \frac{(1-t) \cdot \log n}{\epsilon}\right) \\ &\in \mathcal{O}\left(\left(1 + \frac{(1-t) \cdot \log n}{\epsilon}\right)^2 \cdot \frac{k \cdot d}{\epsilon^{d+1}}\right) \end{aligned}$$

Such  $D$  is an expected set full-filling the requirements of [Theorem 3.1](#). □

### 3.7 Simultaneous coresets for $\ell$ -centrum

Those result also gives a coresets construction for the  $\ell$ -CENTRUM problem. Notice that ORDERED  $k$ -MEDIAN problem instance with a vector  $v$  filled with 1 on first  $\ell$  positions and then 0 on remaining, yields an instance of the unbounded ( $t = 0$ )  $\ell$ -CENTRUM problem.

**Corollary 3.28.** *For any value  $k \in \mathbb{Z}^+$ , data set  $X \subset \mathbb{R}^d$  of size  $|X| = n$ , and any  $\epsilon \in (0, 1)$ , there is a weighted set  $D$  of size  $\mathcal{O}\left(\frac{k^2 \cdot d \cdot \log^2 n}{\epsilon^{d+3}}\right)$  such that:*

$$\text{cost}_\ell(D, C) \in (1 \pm \mathcal{O}(\epsilon)) \cdot \text{cost}_\ell(X, C)$$

for all sets  $C \subset \mathbb{R}^d$  of size  $k$  and  $\ell \leq n$ . Furthermore,  $D$  is computed in polynomial time.

## 4 Fixed-Parameter Tractable approximation algorithm for Ordered $k$ -Median

This chapter combines theorems from the previous section with the framework established by [15], and provides a parameterized approximation algorithm for the ORDERED  $k$ -MEDIAN problem.

The main theorem we will prove is stated as follows:

**Theorem 4.1.** *For every  $\epsilon > 0$ , there is a  $(1 + \frac{2}{1+t} + \mathcal{O}(\epsilon))$ -approximation algorithm for the discrete ORDERED  $k$ -MEDIAN problem with weights vector restricted by  $t$  in the  $\mathbf{L}_p$  norm, that runs in FPT( $k, d$ ) time, i.e., in  $f(k, d, \epsilon) \cdot n^{\mathcal{O}(1)}$  time.*

Moreover, the restriction for  $\mathbf{L}_p$  metric space and the exponential impact of  $d$  is needed only because of the weakness of the coresset construction. Better results (e.g., those matching results from [61] or [53] for  $k$ -MEDIAN) might remove this dependency in further research. Regardless of this fact, non-exponential dependence on the size of the client set should be considered a significant improvement. Additionally, coresets for ORDERED  $k$ -MEDIAN were not intensively explored, so obtaining improved outcomes seems to be highly possible.

### 4.1 Preliminaries

An instance  $I$  of ORDERED  $k$ -MEDIAN is defined as  $(X, F, d, k, v)$ , where  $X$  is the set of clients,  $F$  is the set of facilities. Both clients and facilities are located in a metric space with a distance function  $dist$ .  $k$  is the number of facilities the algorithm should select and  $v$  is the vector of size  $|X|$  representing weights in decreasing order. The precise definition is presented in the beginning of the thesis.

We define the *aspect ratio* of a metric space  $(X \cup F, dist)$  as

$$\Delta := \frac{\max_{v, u \in X \cup F} dist(u, v)}{\min_{v, u \in X \cup F, v \neq u} dist(u, v)}$$

Let  $F^* = \{f_1^*, \dots, f_k^*\}$  be the set of facilities of the optimal solution.

### 4.2 Overview on the construction

**Inspiration** Our construction is built on Cohen et al.'s [15] construction. Similarly, we conduct three steps algorithm including client size reductions, finding *leaders* in the clients' set, and then, using the submodular maximization technique, determining the approximately optimal facilities set.

**Coreset reduction** First, we reduce the set of clients  $C$  to the size  $\mathcal{O}(\frac{k \cdot d \cdot \log n}{e^{d+2}})$  using theorems from the previous section – for the sake of simplicity we use weaker versions of them (not depending on  $t$ ). Instead of considering the original set, we can work with the reduced weighted set whose size is only logarithmically dependent on  $n$ .

**Finding leaders** The next step is enumerating over possible  $k$ -subsets from the reduced  $C$  set to obtain the set of *leaders*  $L = \{l_1, \dots, l_k\}$ . A leader is the closest client to the facility as part of the optimal solution. Also we define  $\forall_{i \in [k]} \text{dist}(l_i, f_i^*) = r_i^*$ . Due to proper multiplicative bucketing, it is possible to determine approximately optimal radii  $R = \{r_1, \dots, r_k\}$  using naive enumeration.

**Determining solutions with approximately submodular function maximization** Moving from the set of optimal leaders to the set of optimal facilities is not straightforward. Picking arbitrary facility  $f_i$ , such that  $\text{dist}(f_i, l_i) = r_i$  gives already  $(3 + \mathcal{O}(\epsilon))$  approximation. However, we observe that selecting facilities in these settings (using the objective of ORDERED  $k$ -MEDIAN instance) can be modeled as an approximately submodular function maximization and employ results from [62] to obtain the desired approximation ratio.

### 4.3 Client set reduction

This part is an apparent application of the [Theorem 3.1](#).

**Corollary 4.2.** *For any value  $k \in \mathbb{Z}^+$ , data set  $X \subset \mathbb{R}^d$  of size  $|X| = n$ , any  $\epsilon \in (0, 1)$  and any  $t \in (0, 1)$ , we can compute in polynomial time a coreset of size  $\mathcal{O}\left(\left(1 + \frac{(1-t) \cdot \log n}{\epsilon}\right)^2 \cdot \frac{k^2 \cdot d}{e^{d+1}}\right)$  correct for every vector of weights  $v \in [t, 1]^n$ .*

In what follows, we assume that the set  $X$  is already reduced with the coreset construction.

### 4.4 $(3 + \mathcal{O}(\epsilon))$ -approximation by finding leaders and radii

The goal of this part is to prove the theorem:

**Lemma 4.3.** *For every  $\epsilon > 0$ , there is a  $(3 + \mathcal{O}(\epsilon))$ -approximation algorithm for the  $t$ -restricted in the  $\mathbf{L}_p$  norm discrete ORDERED  $k$ -MEDIAN problem, that runs in the FPT( $k, d$ ) time, i.e., in  $f(k, d, \epsilon) \cdot n^{\mathcal{O}(1)}$  time.*

*Proof.* As in the overview section, we define a set of (optimal) *leaders* in the following way.

$$L^* = \{\arg \min_{x \in X} \text{dist}(x, f^*) \mid \forall_{f^* \in F^*}\}.$$

We define set  $R^* = \{r_1^*, \dots, r_k^*\}$  in the following way.

$$R^* = \{\min_{x \in X} \text{dist}(x, f^*) \mid f^* \in F^*\}$$

$L^*$  is the (possibly, multi-)subset of  $X$  that contains clients closest to the optimal facilities selection in the given objective. To compute this set, we use a naive approach to iterate the possible selections of clients over  $|X|^k$ .

Additionally, we need to determine  $R = \{r_1, \dots, r_k\}$  being the set of distances such that:

$$\forall_{i \in [k]} r_i \in (1 \pm \epsilon) \cdot r_i^*$$

To get this set, we perform an iteration using multiplicative bucketing. I.e., we initiate iteration with the minimal distance in the metric  $(X \cup F, dist)$  and increase this by  $(1 + \epsilon)$  factor until reaching the maximal distance.

Eventually, for each leader and assigned distance we find any facility such that  $dist(f_i, l_i) \in (1 \pm \epsilon) \cdot r_i$ . Those facilities create set  $F' = \{f_1, \dots, f_k\}$ . The set minimizing the objective function serves as a solution. We will refer to  $f_I(F')$  as ALG.

In pseudo-code, the algorithm works as follows.

Let  $\hat{R}$  be the set of potential radii starting from the minimal distance in the metric and ending at the maximal with  $(1 + \epsilon)$  multiplicative steps. Precisely:

$$\hat{R} = \left\{ \min_{v, u \in X \cup F, v \neq u} dist(u, v) \cdot (1 + \epsilon)^i \mid i \in \{0, 1, \dots, \lfloor \Delta \rfloor\} \right\}$$

We assume that  $f_I(\emptyset) = \infty$ .

---

**Algorithm 1:** Iterative

---

**Data:**  $X, F, \epsilon, v$

**Result:**  $(3 + \mathcal{O}(\epsilon))$ -approximation solution

$F' \leftarrow \emptyset$

**for** every multi-set  $L = \{l_1, \dots, l_k\} \subseteq X$  **do**

**for** every multi-set  $R = \{r_1, \dots, r_k\} \subseteq \hat{R}$  **do**

$\tilde{F} = \{f_i \mid f_i \text{ is any element such that } dist(f_i, l_i) \in (1 \pm \epsilon) \cdot r_i \forall i \in [k]\}$

**if**  $cost_v(F') > cost_v(\tilde{F})$  **then**

$F' \leftarrow \tilde{F}$

**end**

**end**

**end**

**return**  $F'$

---

To conclude the proof, we need to analyze the approximation ratio and the running time.

**Approximation ratio** As argued above, the algorithm correctly determines the following:

- A set of leaders  $L$  that matches the optimal set  $L^*$
- A set of radii  $R$  that is  $(1 + \epsilon)$ -approximately matching  $R^*$ , i.e.,

$$\forall i \in [k] r_i \in (1 \pm \epsilon) \cdot r_i^*$$

We will prove that the for every client  $x \in X$ , whose closest optimal facility is  $f_i^*$ ,  $dist(x, f_i) \leq 3 \cdot dist(x, f_i^*)$  for some  $i \in [k]$ . This will be sufficient to conclude that  $ALG \leq 3 \cdot OPT$ .

First, notice that due to the triangle inequality:

$$\forall_{i \in [k]} \text{dist}(f_i, f_i^*) \leq \text{dist}(f_i, l_i) + \text{dist}(l_i, f_i^*) \leq (2 + 2 \cdot \epsilon) \cdot r_i^*$$

Also, because  $r_i^*$  is a distance to the leader:

$$\forall_{i \in [k]} \text{dist}(x, f_i^*) \geq r_i^*$$

Therefore, due to the triangle inequality:

$$\begin{aligned} \forall_{i \in [k]} \text{dist}(c, f_i) &\leq \text{dist}(x, f_i^*) + \text{dist}(f_i, f_i^*) \\ &\leq \text{dist}(x, f_i^*) + (2 + 2 \cdot \epsilon) \cdot r_i^* \leq (3 + 2 \cdot \epsilon) \cdot \text{dist}(x, f_i^*) \end{aligned}$$

We first assume that  $\forall_{i \in [k]} \text{dist}(x, f_i) = (3 + 2 \cdot \epsilon) \cdot \text{dist}(x, f_i^*)$ . Then  $\text{ALG} = (3 + 2 \cdot \epsilon) \cdot \text{OPT}$ , because the distances are sorted in the same way as in the optimal solution, and each of them is scaled by the same factor  $(3 + 2 \cdot \epsilon)$ .

Eventually, the remaining observation is that every improvement when  $\text{dist}(x, f_i) < (3 + 2 \cdot \epsilon) \cdot \text{dist}(x, f_i^*)$  for some  $i \in [k]$  is improving the objective and the approximation ratio.

**FPT running time** Due to the coreset construction, the number of iterations is bounded as follows:

$$T = \mathcal{O} \left( \left( \left( \left( 1 + \frac{(1-t) \cdot \log n}{\epsilon} \right)^2 \cdot \frac{k^2 \cdot d}{\epsilon^{d+1}} \right)^k \cdot \log^k \Delta \right) \right)$$

Due to the analysis in the appendix of [15],  $\Delta$  is bounded by  $\text{poly}(n)$ . Hence,  $\log \Delta = \mathcal{O}(1) \cdot \log n$ . Therefore:

$$T = \mathcal{O} \left( \left( \left( \left( 1 + \frac{(1-t) \cdot \log n}{\epsilon} \right)^2 \cdot \frac{k^2 \cdot d}{\epsilon^{d+1}} \right)^k \cdot \log^k n \right) \right)$$

First, we assume that  $k \leq \log_{\log n} n$  and  $\frac{(1-t) \cdot \log n}{\epsilon} \leq 1$ . Then:

$$T_1 = \mathcal{O} \left( \left( \left( \frac{k^2 \cdot d}{\epsilon^{d+1}} \right)^k \cdot \log^k n \right) \subseteq \mathcal{O} \left( \left( \frac{k^2 \cdot d}{\epsilon^{d+1}} \right)^k \cdot n \right) \right)$$

Next, we assume that  $k \geq \log_{\log n} n$  and  $\frac{(1-t) \cdot \log n}{\epsilon} \leq 1$ . Then,  $\mathcal{O}(k \cdot \log k) \supseteq \log n$ .

$$T_2 = \mathcal{O} \left( \left( \left( \frac{k^2 \cdot d}{\epsilon^{d+1}} \right)^k \cdot \log^k n \right) \subseteq \mathcal{O} \left( \left( \frac{k^2 \cdot d}{\epsilon^{d+1}} \right)^k \cdot (k \cdot \log k)^k \right) \right)$$

If  $k \leq \log_{\log n} n$  and  $\frac{(1-t) \cdot \log n}{\epsilon} \geq 1$ , then:

$$T_3 = \mathcal{O} \left( \left( \left( \frac{(1-t)^2 \cdot k^2 \cdot d}{\epsilon^{d+3}} \right)^k \cdot \log^{3k} n \right) \subseteq \mathcal{O} \left( \left( \frac{(1-t)^2 \cdot k^2 \cdot d}{\epsilon^{d+3}} \right)^k \cdot n^3 \right) \right)$$

If  $k \geq \log_{\log n} n$  and  $\frac{(1-t) \cdot \log n}{\epsilon} \geq 1$ , then:

$$T_4 = \mathcal{O} \left( \left( \frac{(1-t)^2 \cdot k^2 \cdot d}{\epsilon^{d+3}} \right)^k \cdot \log^{3k} n \right) \subseteq \mathcal{O} \left( \left( \frac{(1-t)^2 \cdot k^2 \cdot d}{\epsilon^{d+3}} \right)^k \cdot (k \cdot \log k)^{3k} \right)$$

Hence, in all cases:

$$T = n^{\mathcal{O}(1)} \cdot f(\epsilon, k, d)$$

That is matching  $\text{FPT}(k, d)$  complexity definition for fixed  $\epsilon$  and concluding the proof of [Lemma 4.3](#). □

Additionally, we observe that the runtime is lower with an increasing value of  $t$ . However, we skip the precise analysis here.

## 4.5 $(1 + \frac{2}{1+t} + \mathcal{O}(\epsilon))$ -approximation by submodular maximization

This part aims to prove [Theorem 4.1](#) eventually. The improvement compared to the results from the previous section is a more careful selection of the set  $F'$ . Previously,  $3 + \mathcal{O}(\epsilon)$  approximation was a consequence of selecting any facility matching the leader and radius in each iteration. The enhanced procedure uses all potential facilities as input for the approximately submodular function. However, we need to ensure that the solution is still feasible. Notably, from the facilities set defined by radius and leader, at most, one may be selected. Otherwise, during optimization, one set like that may be left without any facility remaining, and then, the solution objective will incorrectly consider only the fictitious facility cost. This characteristic requires additional constraint for approximately submodular function optimization, which we model with the matroid structure.

This subsection spits into three parts. The first section presents preliminaries about approximately submodular functions. The next one covers maximization techniques. Eventually, all of those results are put together in the last component.

### 4.5.1 A pproximately submodular function subject to the partition matroid and cardinality constraints

The first definition worth mentioning is the general submodular set function (adapted from [\[63\]](#)).

**Definition 4.4.** *A function  $f : \mathcal{P}(U) \rightarrow \mathbf{R}_+$  is submodular if for all  $S \subseteq T \subseteq U$ ,*

$$\forall_{x \in G} f(T \cup \{x\}) - f(T) \leq f(S \cup \{x\}) - f(S)$$

for a ground set  $U$ .

Strictly or probabilistically limited deviations from the original submodular set function had been a matter of at least a few research efforts in the past (e.g., [64, 65, 66, 67]). However, the direction we are mostly interested in was initiated by Das and Kempe [62] very recently. In this sense, *submodularity ratio* measures the submodularity deviation. For our analysis, we will stick to a more generic definition of *the generic submodularity ratio*.

**Definition 4.5.** [68] and [69] define the generic submodularity ratio  $\gamma(f)$  of a monotone function  $f$  as follows

$$\gamma(f) = \min_{L \subseteq S \subseteq U, x \in U} \frac{f(L \cup \{x\}) - f(L)}{f(S \cup \{x\}) - f(S)}$$

for the given ground set  $U$ .

Notice that  $\gamma(f) = 1$  if the function is submodular.

However, results from the literature (although very promising and yielding an excellent approximation ratio and running time) do not apply to the matroid constraint defined below.

**Definition 4.6.** A matroid is a family  $\mathcal{I}$  of subsets of a ground set  $E$ , satisfying the following:

- $\emptyset \in \mathcal{I}$ ,
- if  $A \in \mathcal{I}$  and then every subset  $B \subseteq A$  is also in  $\mathcal{I}$ , and
- if  $A$  and  $B$  are subsets in  $\mathcal{I}$  with  $|B| > |A|$ , then there exists  $x \in B \setminus A$  such that  $A \cup \{x\}$  is in  $\mathcal{I}$ .

The objective of the submodular maximization subject to the matroid constraint is to ensure that the returned solution stays within the specified matroid. Performing submodular maximization subject to the matroid constraint is essential to design our algorithm. While designing the algorithm, we will enforce the condition that at most 1 facility may be selected from each  $\tilde{F}$  induced by a single leader. Otherwise, some  $\tilde{F}$  might be left with no facility selected, which does not yield a real solution (not using *fictitious* facilities).

The matroid constraint in the perspective of submodular function was initially the subject of interest of Edmonds [70]. Fisher et al. [71] proved that the greedy algorithm yields  $\frac{1}{2}$  approximation.  $(1 - \frac{1}{e})$  approximation was presented by Calinescu [72] using linear optimization. Those are tight due to the results in [73]. Filmus and Ward [74] matched those results by the algorithmic construction. Those results were additionally enhanced to nearly-linear time by Ene and Nguyen [75].

Those two directions - matroid constraints and limited deviation from strict submodularity - need to be merged to serve our purposes. This novel approach was firstly explored by Nong et al [68]. The definition of the ratio was extended by formulating a more flexible understanding of the *generic submodularity ratio*  $\gamma(f)$ . Presented

greedy algorithm yields  $(\frac{\gamma(f)}{1+\gamma(f)})$ -approximation ratio. Those results were improved from the query complexity perspective (however, with the loss of the approximation ratio) by Gong et al. [69] presenting  $(\gamma(f)(1-\frac{1}{e})(1-\frac{1}{e^{\gamma(f)}}))$ -approximation algorithm. Although yielding worse results, this work initiated a research direction continued by the research effort of Liu and Hu obtaining  $(\gamma(f)^2(1-\frac{1}{e})^2)$ -approximation ratio [76], which is still underperforming from the ratio perspective compared to [68]. Those results are summarized in Figure 12.

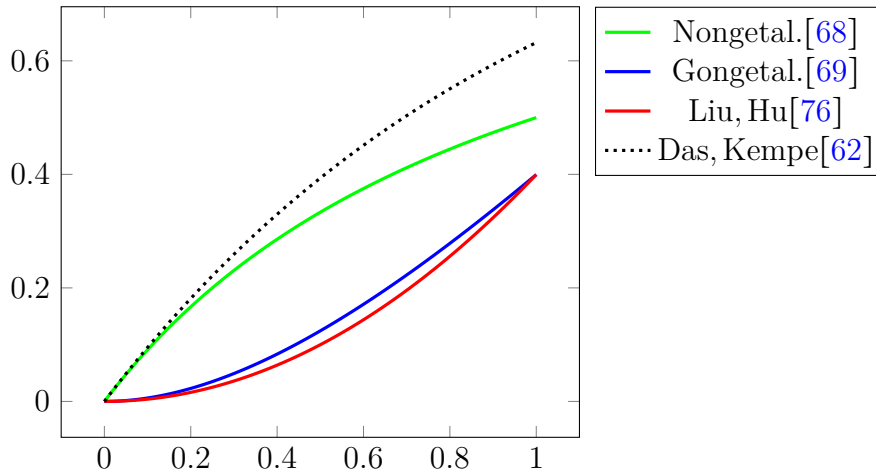


Figure 12: Results of the approximation of the approximately submodular function subject to the matroid constraint with relation to the generic submodularity ratio. Additionally, results of of the approximation of the approximately submodular function without matroid constraint were presented with dotted line as a reference.

Another direction that is related to our approach is the work of Nguyen and Thai [77] exploring the matroid constraint with the limitation to partition matroid, which is sufficient for our needs. However, those results rely on the size of the partition and function curvature and are not applicable in our context.

Together with the definition, the work presented in [68] provides an approximation scheme concluded in the following theorem.

**Theorem 4.7.** *For a monotone set function  $f$ , there exists an  $(\frac{\gamma(f)}{1+\gamma(f)})$ -approximation algorithm determining the set maximizing the function running in polynomial time.*

The proof left in the original paper [68] is a consequence of [78] enriched with careful analysis. We decide to use this algorithm because it offers the best approximation ratio. Obviously, this is the compromise from the computation time perspective.

#### 4.5.2 Modified algorithm and approximately submodularity factor function analysis

The core part of this section is the modification of the previous algorithm. Again, follow the lines of [15]. Firstly, we create a set of *fictitious* facilities that bring worst possible approximation ratio. Then, we improve it with the submodular maximization techniques.

We define a set of *fictitious* facilities  $\hat{F}$ . Those are facilities that are designed to give the worst possible approximation ratio. We will use them to ensure that the function we will provide is monotone. Note that they are not part of  $F$ .

Namely,

$$\hat{F} = \{\hat{f}_i, \dots, \hat{f}_k\}$$

And the distances to facilities considered are maximized:

$$\forall_{f \in \tilde{F}_i} \text{dist}(\hat{f}_i, f) = 2 \cdot r_i$$

Where:

$$\tilde{F}_i = \{f \in F \mid \text{dist}(f, l_i) \in (1 \pm \epsilon) \cdot r_i\}$$

We assume w.l.o.g. that the sets  $\tilde{F}_i$  are disjoint by creating copies and

$$\tilde{F} = \cup_{i \in [k]} \tilde{F}_i$$

To obtain a complete metric, distances to other facilities from fictitious facilities are due to the metric closure. Following similar reasoning as in the previous part, selecting those facilities (assuming their existence) brings  $(3 + \mathcal{O}(1))$ -approximation. Furthermore, selecting additionally other facilities from  $\tilde{F}$  may bring an improvement.

Now, we are ready to define the *improvement* function. This function will be used to improve the selection of facilities with the submodular maximization technique.

$$\text{IMPROV}_{\hat{F},v}(S) = \text{cost}_v(X, \hat{F}) - \text{cost}_v(X, \hat{F} \cup S)$$

**Proposition 4.8.** *For  $k$ -MEDIAN (or for instances of ORDERED  $k$ -MEDIAN where the vector of weights  $v$  satisfies  $v_1 = v_2 = \dots = v_n = 1$ )  $\text{IMPROV}_{\hat{F},v}$  (denoted as  $\text{IMPROV}_{\hat{F},1}$ ) has generic submodularity ratio  $\gamma(\text{IMPROV}_{\hat{F},1}) = 1$ .*

*Proof.* Following the analysis from [15], for  $k$ -MEDIAN is strictly submodular and monotonous. Then,  $\gamma(\text{IMPROV}_{\hat{F},1}) = 1$ .  $\square$

**Lemma 4.9.**  *$\text{IMPROV}_{\hat{F},v}$  is a, approximately submodular function with  $\gamma(\text{IMPROV}_{\hat{F},v}) \geq t$ , i.e., instances of ORDERED  $k$ -MEDIAN such that the vector of weights  $v$  is restricted by  $t$ .*

Be [Definition 4.5](#),

$$\gamma(\text{IMPROV}_{\hat{F},v}) = \min_{L \subseteq S \subseteq \tilde{F}, x \in \tilde{F}} \frac{\text{IMPROV}_{\hat{F},v}(L \cup \{x\}) - \text{IMPROV}_{\hat{F},v}(L)}{\text{IMPROV}_{\hat{F},v}(S \cup \{x\}) - \text{IMPROV}_{\hat{F},v}(S)}$$

Before proving the lemma, we need to show [Proposition 4.10](#) and [Proposition 4.11](#)

Observe that  $\text{IMPROV}_{\hat{F},v}(L \cup \{x\}) - \text{IMPROV}_{\hat{F},v}(L)$  can be considered as a *gain* in the improvement that is induced due to the usage of  $x$ . The *gain* can happen on some clients, and in the one extreme case, this is happening only on clients having the lowest weights in the weights vector (namely,  $t$ ), and in the other extreme, the *gain* happens only with the greatest weights (that is 1).

**Proposition 4.10.**

$$\text{IMPROV}_{\hat{F},v}(L \cup \{x\}) - \text{IMPROV}_{\hat{F},v}(L) \leq \text{IMPROV}_{\hat{F},1}(L \cup \{x\}) - \text{IMPROV}_{\hat{F},1}(L)$$

*Proof.*

$$\begin{aligned} & \text{IMPROV}_{\hat{F},v}(L \cup \{x\}) - \text{IMPROV}_{\hat{F},v}(L) \\ &= (\text{cost}_v(X, \hat{F}) - \text{cost}_v(X, \hat{F} \cup L \cup \{x\})) - (\text{cost}_v(X, \hat{F}) - \text{cost}_v(X, \hat{F} \cup L)) \\ &= \text{cost}_v(X, \hat{F} \cup L) - \text{cost}_v(X, \hat{F} \cup L \cup \{x\}) \end{aligned}$$

Let  $\text{cost}_v^S(C, T)$  denote the objective cost of the ORDERED  $k$ -MEDIAN with selected facilities  $T \in F$ , but with clients sorted in the way as if the set  $S \in F$  was selected. Precisely,

$$\text{cost}_v^S(X, T) = \sum_{i=1}^n v_i \cdot \text{dist}(\arg \max_{x \in X}^i \text{dist}(x, S), T)$$

Notice that such manipulation selection cannot increase the cost because the original assignment of weights and clients is designed to maximize the cost. Therefore,  $\text{cost}_v^S(X, T) \leq \text{cost}_v(X, T)$

Hence,

$$\begin{aligned} & \text{cost}_v(X, \hat{F} \cup L) - \text{cost}_v(X, \hat{F} \cup L \cup \{x\}) \\ & \leq \text{cost}_v(X, \hat{F} \cup L) - \text{cost}_{\hat{F} \cup L}^{\hat{F} \cup L}(X, \hat{F} \cup L \cup \{x\}) \\ &= \sum_{i=1}^n v_i \cdot \max_{x \in X}^i \text{dist}(x, \hat{F} \cup L) - \sum_{i=1}^n v_i \cdot \text{dist}(\arg \max_{x \in X}^i \text{dist}(x, \hat{F} \cup L), \hat{F} \cup L \cup \{x\}) \\ & \leq \sum_{i=1}^n \max_{x \in X}^i \text{dist}(x, \hat{F} \cup L) - \sum_{i=1}^n \text{dist}(\arg \max_{x \in X}^i \text{dist}(x, \hat{F} \cup L), \hat{F} \cup L \cup \{x\}) \\ &= \sum_{x \in X} \text{dist}(x, \hat{F} \cup L) - \sum_{x \in X} \text{dist}(x, \hat{F} \cup L \cup \{x\}) \\ &= \text{cost}_1(X, \hat{F} \cup L) - \text{cost}_1(X, \hat{F} \cup L \cup \{x\}) \\ &= (\text{cost}_1(X, \hat{F}) - \text{cost}_1(X, \hat{F} \cup L \cup \{x\})) - (\text{cost}_1(X, \hat{F}) - \text{cost}_1(X, \hat{F} \cup L)) \\ &= \text{IMPROV}_{\hat{F},1}(L \cup \{x\}) - \text{IMPROV}_{\hat{F},1}(L) \end{aligned}$$

□

**Proposition 4.11.**

$$\text{IMPROV}_{\hat{F},v}(L \cup \{x\}) - \text{IMPROV}_{\hat{F},v}(L) \geq t \cdot (\text{IMPROV}_{\hat{F},1}(L \cup \{x\}) - \text{IMPROV}_{\hat{F},1}(L))$$

*Proof.* The proof resembles conceptually the proof the [Proposition 4.10](#)

$$\begin{aligned}
& \text{IMPROV}_{\hat{F},v}(L \cup \{x\}) - \text{IMPROV}_{\hat{F},v}(L) \\
&= \text{cost}_v(X, \hat{F} \cup L) - \text{cost}_v(X, \hat{F} \cup L \cup \{x\}) \\
&\geq \text{cost}_v^{\hat{F} \cup L \cup \{x\}}(X, \hat{F} \cup L) - \text{cost}_v(X, \hat{F} \cup L \cup \{x\}) \\
&= \sum_{i=1}^n v_i \cdot \text{dist}(\arg \max_{x \in X}^i \text{dist}(x, \hat{F} \cup L \cup \{x\}), \hat{F} \cup L) - \sum_{i=1}^n v_i \cdot \max_{x \in X}^i \text{dist}(x, \hat{F} \cup L \cup \{x\}) \\
&\geq \sum_{i=1}^n t \cdot \text{dist}(\arg \max_{x \in X}^i \text{dist}(x, \hat{F} \cup L \cup \{x\}), \hat{F} \cup L) - \sum_{i=1}^n t \cdot \max_{x \in X}^i \text{dist}(x, \hat{F} \cup L \cup \{x\}) \\
&= t \cdot (\text{cost}_1(X, \hat{F} \cup L) - \text{cost}_1(X, \hat{F} \cup L \cup \{x\})) \\
&= t \cdot (\text{IMPROV}_{\hat{F},1}(L \cup \{x\}) - \text{IMPROV}_{\hat{F},1}(L))
\end{aligned}$$

□

*Proof of Lemma 4.9.* Because of [Proposition 4.10](#) and [Proposition 4.11](#):

$$\gamma(\text{IMPROV}_{\hat{F},v}) \geq \min_{L \subseteq S \subseteq \tilde{F}, x \in \tilde{F}} \frac{t \cdot (\text{IMPROV}_{\hat{F},1}(L \cup \{x\}) - \text{IMPROV}_{\hat{F},1}(L))}{\text{IMPROV}_{\hat{F},1}(S \cup \{x\}) - \text{IMPROV}_{\hat{F},1}(S)}$$

Consequently,

$$\gamma(\text{IMPROV}_{\hat{F},v}) \geq t \cdot \left( \min_{L \subseteq S \subseteq \tilde{F}, x \in \tilde{F}} \frac{\text{IMPROV}_{\hat{F},1}(L \cup \{x\}) - \text{IMPROV}_{\hat{F},1}(L)}{\text{IMPROV}_{\hat{F},1}(S \cup \{x\}) - \text{IMPROV}_{\hat{F},1}(S)} \right)$$

Next, we substitute the submodularity ratio for  $\text{IMPROV}_{\hat{F},1}$  and use [Proposition 4.8](#),

$$\gamma(\text{IMPROV}_{\hat{F},v}) \geq t \cdot \gamma(\text{IMPROV}_{\hat{F},1}) = t$$

□

Now, we are ready to present the full  $\text{FPT}(k, d)$  algorithm with approximately submodular maximization (referred in the pseudo-code as simply *maximization*). For the same of simplicity, in the pseudo-code, we omit the description of the *maximization* subroutine. The objective of this part, as discussed earlier, to carefully select the facilities yielding the greatest *improvement* compared to the original selection of the *fictitious* facilities. Additionally, we employ matroid constraint to assure that from each  $\tilde{F}_i$  only one facility will be selected.

*Proof of Theorem 4.1.* To show the running time of [Algorithm 2](#), this is enough to observe that the submodular maximization in the inner loop runs in polynomial time. Hence the analysis from the previous section implies FPT running time. The approximation ratio needs more careful analysis.

First, we assume that  $F'$  (being the result of the computation done by the algorithm) has size  $k$ . That means that no fictitious facility has been used during

---

**Algorithm 2:** Iterative with approximately submodular maximization
 

---

**Data:**  $X, F, \epsilon, v$   
**Result:**  $(3 + \mathcal{O}(\epsilon))$ -approximation solution  
 $F' \leftarrow \emptyset$   
**for** every multi-set  $L = \{l_1, \dots, l_k\} \subseteq X$  **do**  
     **for** every multi-set  $R = \{r_1, \dots, r_k\} \subseteq \hat{R}$  **do**  
          $\tilde{F} \leftarrow \emptyset$   
         **for**  $i \in \{1, \dots, k\}$  **do**  
              $\tilde{F}_i \leftarrow \{f \in F \mid \text{dist}(f, l_i) \in (1 \pm \epsilon) \cdot r_i\}$   
              $\tilde{F} \leftarrow \tilde{F} \cup \tilde{F}_i$   
         **end**  
          $\hat{F} \leftarrow$  set of fictitious facilities  
          $\bar{F} \leftarrow$  result of maximization of  $\text{IMPROV}_{\hat{F}, v}(S)$  for  $S \in \tilde{F}$  and  $k$   
         **if**  $\text{cost}_v(F') > \text{cost}_v(\bar{F})$  **then**  
              $F' \leftarrow \bar{F}$   
         **end**  
     **end**  
**end**  
**return**  $F'$

---

computing the cost and  $\text{cost}_v(F') = \text{cost}_v(F' \cup \hat{F})$ . If this is not true, that means that some  $\tilde{F}_i$  was not used, and we can greedily include any facility from this set, yielding another solution satisfying cardinality and matroid constraint without decreasing the *improvement*.

Therefore,

$$\begin{aligned}
 \text{cost}_v(X, F') &= \text{cost}_v(X, \hat{F} \cup F') \\
 &= \text{cost}_v(X, \hat{F} \cup F') + \text{cost}_v(X, \hat{F}) - \text{cost}_v(X, \hat{F}) \\
 &= \text{cost}_v(X, \hat{F}) - \text{IMPROV}_{\hat{F}, v}(F')
 \end{aligned}$$

Due to [68]  $\text{IMPROV}_{\hat{F}, v}$  function was approximated with  $(\frac{t}{1+t})$  ratio. Therefore,

$$\begin{aligned}
 \text{cost}_v(X, F') &\leq \text{cost}_v(X, \hat{F}) - \left(\frac{t}{1+t}\right) \cdot \text{IMPROV}_{\hat{F}, v}(F^*) \\
 &= \text{cost}_v(X, \hat{F}) - \left(\frac{t}{1+t}\right) \cdot (\text{cost}_v(X, \hat{F}) - \text{cost}_v(X, F^*)) \\
 &= \left(\frac{1}{1+t}\right) \cdot \text{cost}_v(X, \hat{F}) + \left(\frac{t}{1+t}\right) \cdot \text{cost}_v(X, F^*)
 \end{aligned}$$

Additionally, *fictitious* facilities yield  $(3 + \mathcal{O}(\epsilon))$ -approximation ratio due to the analysis done in the previous section.

Therefore,

$$\begin{aligned} \text{cost}_v(X, F') &\leq (3 + \mathcal{O}(\epsilon)) \cdot \left(\frac{1}{1+t}\right) \cdot \text{cost}_v(X, F^*) + \left(\frac{t}{1+t}\right) \cdot \text{cost}_v(X, F^*) \\ &= \left(\frac{3+t+\mathcal{O}(\epsilon)}{1+t}\right) \cdot \text{cost}_v(X, F^*) \leq \left(1 + \frac{2}{1+t} + \mathcal{O}(\epsilon)\right) \cdot \text{cost}_v(X, F^*) \end{aligned}$$

□

## 4.6 A naive approach for large $t$

Due to the weaker results related to the maximizing approximately submodular function, the results do not smoothly match the ratio obtained [15] for  $k$ -MEDIAN, i.e., for  $t = 1$ . Because of this, we present a naive approach to simply reuse the solution yielding  $(1 + \frac{2}{e} + \mathcal{O}(\epsilon))$ -approximation ratio for  $k$ -MEDIAN and estimate how far in the objective value is the optimal solution for  $k$ -MEDIAN used in ORDERED  $k$ -MEDIAN instance from the optimal solution for the ORDERED  $k$ -MEDIAN instance.

**Theorem 4.12.** *There exists a FPT( $k$ )  $(\frac{e+2}{t \cdot e} + \mathcal{O}(\epsilon))$ -approximation for ORDERED  $k$ -MEDIAN in the general metric space for instances of ORDERED  $k$ -MEDIAN with the vector of weights  $v$  restricted by  $t$ .*

*Proof.* Let  $\text{OPT}_{\text{med}}$  denote the value of the optimal solution for  $k$ -MEDIAN. Let  $(\text{OPT}_{\text{med}})_v$  denote the value of optimal solution for  $k$ -MEDIAN under the ORDERED  $k$ -MEDIAN objective with the vector of weights  $v$  restricted by the parameter  $t$ . Let  $\text{OPT}_v$  denote the optimal solution for that ORDERED  $k$ -MEDIAN instance.  $(\text{OPT}_{\text{med}})_{v-t}$  is the partial objective of  $(\text{OPT}_{\text{med}})_v$  calculated with the assumption that every element of the vector of weights  $v$  is decreased by  $t$ . Similarly,  $(\text{OPT}_{\text{med}})_t = (\text{OPT}_{\text{med}})_v - (\text{OPT}_{\text{med}})_{v-t}$ . Then,  $(\text{OPT}_{\text{med}})_{v-t} \leq \frac{1-t}{t} \cdot (\text{OPT}_{\text{med}})_t$ .

Therefore,

$$(\text{OPT}_{\text{med}})_v = (\text{OPT}_{\text{med}})_t + (\text{OPT}_{\text{med}})_{v-t} \leq (\text{OPT}_{\text{med}})_t + \frac{1-t}{t} \cdot (\text{OPT}_{\text{med}})_t = \frac{(\text{OPT}_{\text{med}})_t}{t}$$

In the same way we define  $(\text{OPT}_v)_{v-t}$  and  $(\text{OPT}_v)_t$ . Obviously,  $(\text{OPT}_v)_{v-t} + (\text{OPT}_v)_t = \text{OPT}_v$  and  $(\text{OPT}_{\text{med}})_t < (\text{OPT}_v)_t$ , because this part of the objective is actually an instance of  $k$ -MEDIAN problem that is minimized in the optimum.

Hence,

$$(\text{OPT}_{\text{med}})_v \leq \frac{(\text{OPT}_v)_t}{t} \leq \frac{(\text{OPT}_v)_t + (\text{OPT}_v)_{v-t}}{t} = \frac{\text{OPT}_v}{t}$$

Thereby, using the  $(1 + \frac{2}{e} + \mathcal{O}(\epsilon))$ -approximation algorithm from [15], we obtain the a desired approximation ratio.

$$\left(1 + \frac{2}{e} + \mathcal{O}(\epsilon)\right) \cdot \frac{1}{t} \leq \frac{e+2}{t \cdot e} + \mathcal{O}(\epsilon)$$

□

## 4.7 Summary

Figure 13 concludes the results from the previous sections. We observe that  $t = \frac{1-e+\sqrt{1+2\cdot e^2}}{e} (\approx 0.83)$  is a point after which the naive solution yields a better approximation ratio. However, the naive solution is not dependent on the coresset construction and, as in work by [15], can be used in the general metric space. Notably, in the case of Euclidean space, the running time does not incur any dependency on the dimensions.

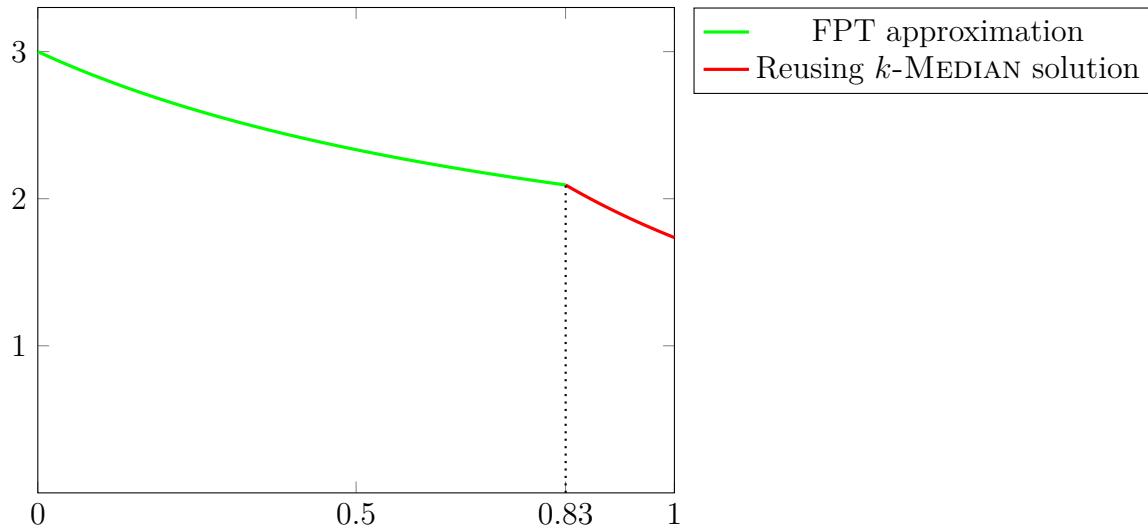


Figure 13: Summary of the approximation results from this chapter with respect to the parameter  $t$ .

## 5 Hardness of $\ell$ -centrum and Ordered $k$ -Median approximation

Our results are essential for understanding the possible approximation factors yielded by a polynomial and fixed-polynomial time algorithms. However, we can also derive opposite results showing inapproximability up a given factor under the standard assumptions, i.e.,  $\mathbf{P} \neq \mathbf{NP}$ .

Results in this chapter are narrowed to a specific reduction to the SET-COVER problem that is known to be  $\mathbf{NP}$ -hard inspired by [27]. Also, we do not consider continuous variants of the problems limiting ourselves to instances with a specified set of facilities (the other group has already been analyzed in the literature, e.g., [79] and [80] for  $\ell$ -CENTRUM). This section iterates over  $k$ -CENTRUM and  $\ell$ -CENTRUM problems and eventually concludes with results for ORDERED  $k$ -MEDIAN.

In the following deduction, we will ignore the  $d$  factor. Thus, we will focus on the general metric case. A decent amount of work has already been devoted to the Euclidean space, particularly for the  $k$ -MEDIAN problem, i.e., [81], or  $\ell$ -CENTRUM, i.e., [82].

The following deductions are implicitly assuming that  $\mathbf{NP} \neq \mathbf{P}$ . That means that finding algorithms satisfying those hardness requirements is breaking reasonable and widely-believed assumptions about not collapsing those complexity classes.

### 5.1 Hardness of $k$ -Centrum approximation

**Theorem 5.1.** *If there exists a polynomial-time algorithm solving the  $k$ -CENTRUM problem within  $(3 - \epsilon)$  approximation factor, then  $\mathbf{NP} = \mathbf{P}$ .*

The proof shows a reduction of the decisive version of SET-COVER problem instance known to be  $\mathbf{NP}$ -complete [83] to the  $k$ -CENTRUM problem instance.

Definition of an instance  $I_{SC}$  of the SET-COVER decisive version includes:

- The ground set  $U$ ,
- Subsets  $\mathcal{S}$ , such that  $\forall_{S \in \mathcal{S}} S \subseteq U$ ,
- $h \in \mathbb{N}$  indicating the allowed number of subsets.

The decisive version of the SET-COVER problem answers the question of whether this is possible to cover  $U$  with subset  $\mathcal{S}' \subseteq \mathcal{S}$  such that  $|\mathcal{S}'| \leq k$ .

**Lemma 5.2.** *There exists a polynomial-time algorithm transforming the SET-COVER problem instance  $I_{SC}$  into the  $k$ -CENTRUM problem instance  $I$  such that:*

- *If  $I_{SC}$  is feasible, then  $\text{OPT}_I = 1$*
- *If  $I_{SC}$  is not feasible, then  $\text{OPT}_I \geq 3$*

*Proof.* With the following reduction, we obtain the instance  $I = (X, F, M, k)$  of the  $k$ -CENTRUM problem from any  $I_{SC} = (U, \mathcal{S}, k)$ .

For each  $S \in \mathcal{S}$ , let  $f_S$  be a facility derived from this subset. For  $v \in U$ , let  $c_v$  be a client. Then,  $F = \{f_S : S \in \mathcal{S}\}$  and  $X = \{c_v : v \in U\}$ . Also, we define  $c_u^{-1} = v$  if  $c_v = u$

and  $f_m^{-1} = S$  if  $f_S = m$ . To obtain a metric space, firstly let  $dist(u, v) = 1$  if  $u \in F$ ,  $v \in X$  (or other way) and  $c_v^{-1} \in f_u^{-1}$ . The bipartite graph formed this way obtains the complete metric space via the shortest path relaxation.

**Observation 5.3.** *After the relaxation, the following holds:*

$$dist(u, v) = \begin{cases} 1 & \text{if } u \in F, v \in X \text{ (or other way) and } c_v^{-1} \in f_u^{-1} \\ \geq 3 & \text{if } u \in F, v \in X \text{ (or other way) and } c_v^{-1} \notin f_u^{-1} \\ \geq 2 & \text{otherwise} \end{cases}$$

Also, let  $k = h$  and  $M = (X \cup F, dist)$  be metric space. As a result, we have a complete instance of the  $k$ -CENTRUM problem. Figure 14 shows the example of the projection of the decisive SET-COVER instance to the metric space.

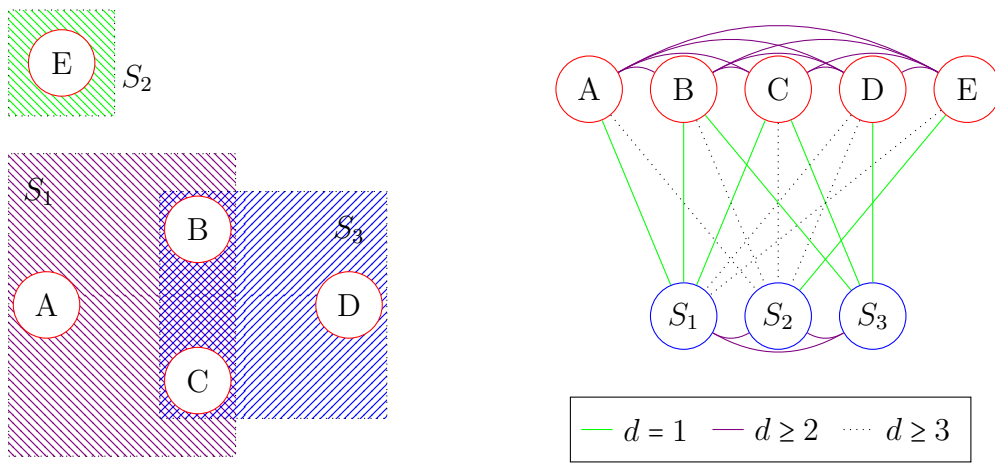


Figure 14: Example of the projection of the SET-COVER problem instance on the left side to the metric space on the right side.

Observe that if this is possible to cover  $U$  from  $I_{SC}$  with  $h$  sets, then this is possible to select  $k$  facilities such that all clients have a distance one to the closes facility, so  $OPT_I = 1$ . Otherwise, using [Observation 5.3](#), at least one client had distance at least 3, meaning that  $OPT_I \geq 3$ .

This finishes the proof of [Lemma 5.2](#) □

*Proof of [Theorem 5.1](#).* The obvious corollary from [Lemma 5.2](#) is that any efficient algorithm solving the  $k$ -CENTRUM problem with  $(3 - \epsilon)$ -approximation ratio can be used to yield the answer for the decisive (and therefore, general) SET-COVER instance in polynomial time. Then,  $\mathbf{P} = \mathbf{NP}$ , which concluded the proof of [Theorem 5.1](#). □

## 5.2 The hardness of $\ell$ -centrum approximation with respect to $\ell$

**Theorem 5.4.** *There is no polynomial time algorithm solving the  $\ell$ -CENTRUM problem instance within  $\min(3, 1 + \frac{2n}{e\ell}) - \epsilon$  approximation factor, where  $n$  is the number of clients, unless  $\mathbf{P} = \mathbf{NP}$ .*

To prove the theorem above, we need the two ingredients. The first is related to the hardness constraints when  $\ell \leq \frac{n}{e}$  and the complementary part when  $\ell > \frac{n}{e}$ .  $h(\ell) = \min(3, 1 + \frac{2 \cdot n}{e \cdot \ell})$  curve is composed from two elementary function. When  $\ell \leq \frac{n}{e}$ ,  $3 \leq 1 + \frac{2 \cdot n}{e \cdot \ell}$ . Otherwise, the opposite of inequality holds.

Hence, this is enough to prove the following lemmas. Due to the manipulation of  $\epsilon$ , we obtain negligibly stronger results, which are used to conclude [Theorem 5.4](#).

**Lemma 5.5.** *There is no polynomial time algorithm yielding  $3 - \epsilon$  approximation factor solving the  $\ell$ -CENTRUM problem with  $\ell < (\frac{1}{e} - \frac{\epsilon}{6}) \cdot n$ , unless  $\mathbf{P} = \mathbf{NP}$ .*

**Lemma 5.6.** *There is no polynomial time algorithm yielding  $1 + \frac{2 \cdot n}{e \cdot \ell} - \epsilon$  approximation factor solving the  $\ell$ -CENTRUM problem with  $\ell \geq (\frac{1}{e} - \frac{\epsilon}{6}) \cdot n$ , unless  $\mathbf{P} = \mathbf{NP}$ .*

Before proving any of those, we need an additional fact: the implication of results from [\[84\]](#) about MAX- $k$ -COVER and SET-COVER considering their decisive versions. This is already well-known and proven in the thesis of Richard Karp [\[83\]](#) that a polynomial-time algorithm deciding whether this is possible to cover the ground set with  $k$  sets causes  $\mathbf{NP}$  complexity class collapse. However, Uriel Feige provides even strong results:

**Corollary 5.7.** *Unless  $\mathbf{P} = \mathbf{NP}$ , there is no polynomial time algorithm distinguishing following cases for the given SET-COVER instance  $I_{SC} = (U, \mathcal{S}, h)$ :*

1. *Either, ground set  $U$  is fully-covered with some  $h$  subsets from  $\mathcal{S}$ ,*
2. *Or, there is no  $h$  subset of subsets from  $\mathcal{S}$  covering  $(1 - \frac{1}{e} + \hat{\epsilon})$  part of  $U$  for any positive  $\hat{\epsilon}$ .*

We use  $\hat{\epsilon}$  to distinguish it from  $\epsilon$  because they might not be equal. In the following deductions, w.l.o.g. we assume that  $\hat{\epsilon} = \frac{\epsilon}{6}$ .

Now, we are ready to prove those lemmas. For both, we will use a similar strategy as in [Theorem 5.1](#) showing the following reductions.

**Lemma 5.8.** *There exists a polynomial-time algorithm transforming the SET-COVER problem instance  $I_{SC}$  into the  $\ell$ -CENTRUM problem instance  $I_1$  for  $\ell < n \cdot (\frac{1}{e} - \frac{\epsilon}{6})$  such that:*

- *If the ground set  $U$  is fully-covered with some  $h$  subsets from  $\mathcal{S}$ , then  $\text{OPT}_{I_1} = \ell$*
- *If there is no  $h$  subset of subsets from  $\mathcal{S}$  covering  $(1 - \frac{1}{e} + \hat{\epsilon})$  part of  $U$ , then  $\text{OPT}_{I_1} > \ell \cdot (3 - \epsilon)$*

And analogically,

**Lemma 5.9.** *There exists a polynomial-time algorithm transforming the SET-COVER problem instance  $I_{SC}$  into the  $\ell$ -CENTRUM problem instance  $I_2$  for  $\ell \geq n \cdot (\frac{1}{e} - \frac{\epsilon}{6})$  such that:*

- *If the ground set  $U$  is fully-covered with some  $h$  subsets from  $\mathcal{S}$ , then  $\text{OPT}_{I_2} = \ell$*
- *If there is no  $h$  subset of subsets from  $\mathcal{S}$  covering  $(1 - \frac{1}{e} + \hat{\epsilon})$  part of  $U$ , then  $\text{OPT}_{I_2} > \ell \cdot (1 + \frac{2 \cdot n}{e \cdot \ell} - \epsilon)$*

For both lemmas, the reduction is exactly the same as in [Theorem 5.1](#), the difference lies only within the objective function used depending on the selection of  $\ell$  value.

*Proof of [Lemma 5.8](#).* If the ground set  $U$  is fully covered, then  $\text{OPT}_{I_1} = \ell$ , because choosing the facilities corresponding to the optimal set covering yields the best possible solution.

In the second case,  $\text{OPT}_{I_1} \geq 3 \cdot \ell$ , because none of the clients within the first most remote  $\frac{1}{e} - \hat{\epsilon} = \frac{1}{e} - \frac{\epsilon}{6}$  has distance 1 to any facilities. Otherwise, this will lead to a contradiction cause this facility selection would correspond to a subset of sets covering more  $(1 - \frac{1}{e} + \hat{\epsilon})$  fraction of  $U$ . Hence, then  $\text{OPT}_{I_1} > (3 - \epsilon) \cdot \ell$   $\square$

*Proof of [Lemma 5.5](#).* The corollary from [Lemma 5.8](#) is that any efficient solving  $\ell$ -CENTRUM problem for  $\ell < n \cdot (\frac{1}{e} - \frac{\epsilon}{6})$  with  $(3 - \epsilon)$ -approximation ratio can be used to yield the answer for the decisive SET-COVER instance in polynomial time. Then,  $\mathbf{P} = \mathbf{NP}$ , which concluded the proof  $\square$

*Proof of [Lemma 5.9](#).* If the ground set  $U$  is fully covered, then, similarly,  $\text{OPT}_{I_2} = \ell$ .

In the second case,  $\text{OPT}_{I_1} \geq (\ell - (\frac{1}{e} + \hat{\epsilon}) \cdot n) + 3 \cdot (\frac{1}{e} - \hat{\epsilon}) \cdot n$ , because at least  $(\frac{1}{e} - \hat{\epsilon})$  clients needs to have the other weight at least 3 and all weights needs to be least 1.

Hence,

$$\text{OPT}_{I_1} \geq \ell + 2 \cdot (\frac{1}{e} - \hat{\epsilon}) \cdot n = \ell \cdot (1 + \frac{2 \cdot n}{e \cdot \ell} - \frac{\epsilon \cdot n}{6 \cdot \ell})$$

Observe that since  $\epsilon \leq 1$  and  $\ell \geq n \cdot (\frac{1}{e} - \frac{\epsilon}{6})$ ,  $\ell > 0.2 \cdot n$ . Hence,

$$\frac{\epsilon \cdot n}{6 \cdot \ell} \leq \frac{\epsilon \cdot n}{6 \cdot 0.2 \cdot n}$$

Therefore,

$$\text{OPT}_{I_1} > \ell \cdot (1 + \frac{2 \cdot n}{e \cdot \ell} - \frac{\epsilon \cdot n}{6 \cdot 0.2 \cdot n}) > \ell \cdot (1 + \frac{2 \cdot n}{e \cdot \ell} - \epsilon)$$

$\square$

*Proof of [Lemma 5.6](#).* The corollary from [Lemma 5.9](#) is that any efficient algorithm solving  $\ell$ -CENTRUM problem for  $\ell \geq n \cdot (\frac{1}{e} - \frac{\epsilon}{6})$  with  $(1 + \frac{2 \cdot n}{e \cdot \ell} - \epsilon)$ -approximation ratio can be used to yield the answer for the decisive SET-COVER instance in polynomial time. Then,  $\mathbf{P} = \mathbf{NP}$ , which concluded the proof.  $\square$

Eventually,

*Proof of [Theorem 5.4](#).* [Lemma 5.5](#) and [Lemma 5.6](#) together conclude the proof of [Theorem 5.4](#). Namely,

1. For  $0 < \ell < n \cdot (\frac{1}{e} - \frac{\epsilon}{6})$ , the hardness is obtained using [Lemma 5.5](#).
2. For  $n \cdot (\frac{1}{e} - \frac{\epsilon}{6}) \leq \ell < \frac{n}{e}$ , the hardness is obtained using [Lemma 5.6](#). However, for the sake of simplicity, we observe that  $1 + \frac{2 \cdot n}{e \cdot \ell} - \epsilon > 3 - \epsilon$  for those value of  $\ell$  and any positive  $\epsilon$ .

3. For  $\frac{n}{e} \leq \ell \leq n$ , the hardness is obtained using [Lemma 5.6](#).

□

Note that this is matching tight results for  $k$ -MEDIAN previously stated in [27] what is an additional argument for the non-triviality of this bound. Although we do not provide any results about tightness in the general  $\ell$ -CENTRUM settings.

### 5.3 The hardness of Ordered $k$ -Median approximation with respect to the parameter $t$

Although results from the previous subsection are new, they are not particularly interesting while considering the ORDERED  $k$ -MEDIAN problem. Obviously, we can observe that  $\ell$ -CENTRUM is a special case of ORDERED  $k$ -MEDIAN. Then subproblems of ORDERED  $k$ -MEDIAN with restricted vector by a value  $\ell$  that is a number of non-zero elements in a weights vector are having at least the same hardness as stated in [Theorem 5.4](#).

However, those results are not important in our current deductions, i.e., with respect to the ratio of the smallest and the greatest values of the weight vector. Observe, that if we start restricting the vector by limiting the number of non-zero elements, immediately the parameter  $t$  is set to 0. Thereby, we still do not possess any information about hardness except for the  $1 + \frac{2}{e}$  hardness. However, this is the obvious implication even from [27] and the observation that  $k$ -MEDIAN is a special case of ORDERED  $k$ -MEDIAN.

The goal is to obtain a non-trivial hardness lower bound parameterized by the parameter  $t$ .

**Theorem 5.10.** *Unless  $\mathbf{P} = \mathbf{NP}$ , there is no polynomial-time approximation algorithm solving the ORDERED  $k$ -MEDIAN instance within  $h(t) - \epsilon$  approximation factor, where  $h(t)$  is defined in the following way*

$$h(t) = \begin{cases} \frac{3}{1+(e-1)t} & \text{if } t < \frac{2}{2+e} \\ 1 + \frac{2}{e} & \text{if } t \geq \frac{2}{2+e} \end{cases}$$

The approach used here is inspired by the objective function of the  $k$ -CENTDIAN problem introduced in [46]. The  $k$ -CENTDIAN cost function is a convex combination of functions applied in  $k$ -MEDIAN and  $\ell$ -CENTRUM. This multi-objective vector is filled with weights of values 1 in the initial  $\ell$  entries and the remaining have assigned value  $t$ . Hence, the instance of the  $k$ -CENTDIAN problem is defined as  $I_{\ell,t} = (C, F, k, \ell, t)$ . Observe that such an instance is a special case of the ORDERED  $k$ -MEDIAN instance restricted with  $t$ . Therefore, the hardness of the  $k$ -CENTDIAN problem parameterized by  $t$  is also the hardness of the ORDERED  $k$ -MEDIAN problem parameterized by the same value of  $t$ .

To state the proof, we need those the following lemmas. Then, via contradiction, we will show that the algorithm yielding the following approximation ratio cannot exist due to the efficient reduction to  $\ell$ -CENTRUM.

**Lemma 5.11.** *There exists a polynomial-time reduction transforming the instance  $I_\ell$  of the  $\ell$ -CENTRUM problem into the instance  $I_{\ell,t}$  of the  $k$ -CENTDIAN problem such that  $\text{OPT}_{I_{\ell,t}} \leq (1 + \frac{n-\ell}{\ell} \cdot t) \cdot \text{OPT}_{I_\ell}$*

*Proof.* The procedure is to simply use the selected value  $t$  and use the same selection of  $\ell$  while obtaining  $I_{\ell,t}$  from  $I_\ell$ .

To prove the optimal solutions bound, this is enough to observe the vector weight of the  $k$ -CENTDIAN problem instance. Since the weights are decreasing together with weights, the sum of the weighted distances with weight  $t$  must be less or equal to  $\frac{n-\ell}{\ell} \cdot t$ . Additionally, the solution obtained this way might be sub-optimal, strengthening the theorem.  $\square$

Figure 15 presents the weights vector for the  $k$ -CENTDIAN instance. Figure 16 shows the weighted distances vector that supports understanding of the inequality above.

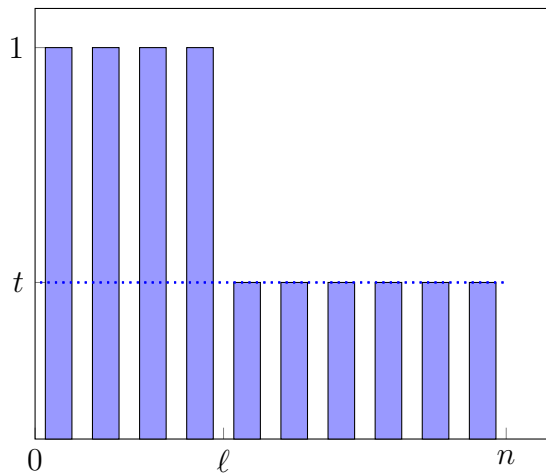


Figure 15: Vector of the weights in the  $k$ -CENTDIAN problem instance for some  $\ell$  and  $\alpha$  divided into two parts. The first one is matching the vector of the  $\ell$ -CENTRUM problem instance. The second one is equal to  $\frac{n-\ell}{\ell} \cdot t$  part of the first one.

**Lemma 5.12.** *There exists a polynomial-time reduction transforming the solution of the instance  $I_{\ell,t}$  of the  $k$ -CENTDIAN problem into a solution of the instance  $I_\ell$  of the  $\ell$ -CENTRUM problem such that  $\text{ALG}_{I_{\ell,t}} \leq \text{ALG}_{I_\ell}$ .*

*Proof.* The procedure is to use the same solution and observe that the objective of any solution of  $I_{\ell,t}$  will be greater than or equal to the objective of the same solution used for  $I_\ell$  for the same selection of clients and facilities.  $\square$

Eventually,

*Proof of Theorem 5.10.* Now, to prove Theorem 5.10 we assume toward contradiction the existence of the algorithm solving  $I_{\ell,t}$  with the approximation ratio  $\frac{\min(3, 1 + \frac{2 \cdot n}{e \cdot \ell})}{1 + \frac{n-\ell}{\ell} \cdot t} - \epsilon$ .

Using Lemma 5.11 and Lemma 5.12, the observation is that this algorithm can be used to yield the solution of  $\ell$ -CENTRUM with the following approximation ratio.

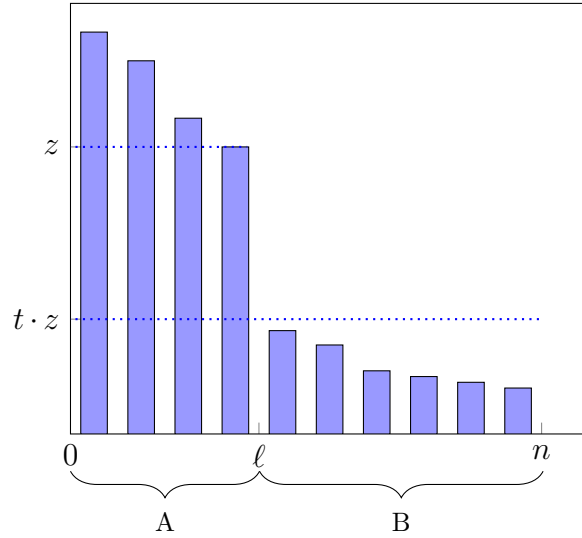


Figure 16: Vector of the weighed distances for some solution of  $k$ -CENTDIAN. Part  $A$  can be lower-bounded by  $z \cdot \ell$ . Part  $B$  is upper-bounded by  $k \cdot (n - \ell)$ . Assuming we used the optimal solution for  $\ell$ -CENTRUM to solve the  $k$ -CENTDIAN instance, the objective solution for this  $k$ -CENTDIAN instance is smaller than  $A + B$ .

$$\left(\frac{\min(3, 1 + \frac{2n}{e\ell})}{1 + \frac{n-\ell}{\ell} \cdot t} - \epsilon\right) \cdot \left(1 + \frac{n-\ell}{\ell} \cdot t\right) \leq \min(3, 1 + \frac{2 \cdot n}{e \cdot \ell}) - \epsilon$$

This is the clear contradiction with [Theorem 5.4](#) and implies  $\mathbf{P} = \mathbf{NP}$ . Therefore,

**Corollary 5.13.** *There is no polynomial-time approximation algorithm solving the  $k$ -CENTDIAN instance  $I_{\ell,t}$  within  $\frac{\min(3, 1 + \frac{2n}{e\ell})}{1 + \frac{n-\ell}{\ell} \cdot t} - \epsilon$  approximation factor, unless  $\mathbf{P} = \mathbf{NP}$ .*

The  $k$ -CENTDIAN instance can be considered a special case of ORDERED  $k$ -MEDIAN restricted by  $y$ . Hence, using [Corollary 5.13](#), the hardness of  $k$ -CENTDIAN (and therefore ORDERED  $k$ -MEDIAN) depending on  $t$  (denoted as  $h(t)$ ) without considering  $\ell$  may be described as:

$$h(t) = \max_{0 \leq \ell \leq n} \frac{\min(3, 1 + \frac{2n}{e\ell})}{1 + \frac{n-\ell}{\ell} \cdot t} - \epsilon$$

Let  $h_t(\ell) = \frac{\min(3, 1 + \frac{2n}{e\ell})}{1 + \frac{n-\ell}{\ell} \cdot t} - \epsilon$ . We observe that  $h_t(\ell)$  is monotone for fixed  $t \in [0, 1]$  over ranges  $[0, \frac{n}{e}]$  and  $[\frac{n}{e}, n]$  of  $\ell$ . Also, for any  $t$ ,  $h_t(0) \leq h_t(\frac{n}{e})$ .

Thus, we can conclude the results about hardness in the following way:

$$h(t) = \max(h_{\frac{n}{e}}(t), h_n(t))$$

Where:

$$h_{\frac{n}{e}}(t) = \frac{3}{1 + \frac{n-\frac{n}{e}}{\frac{n}{e}} \cdot t} = \frac{3}{1 + (e-1) \cdot t} - \epsilon$$

$$h_n(t) = 1 + \frac{2}{e} - \epsilon$$

Figure 17 shows those two curves  $-h_{\frac{n}{e}}$  and  $h_n$ .

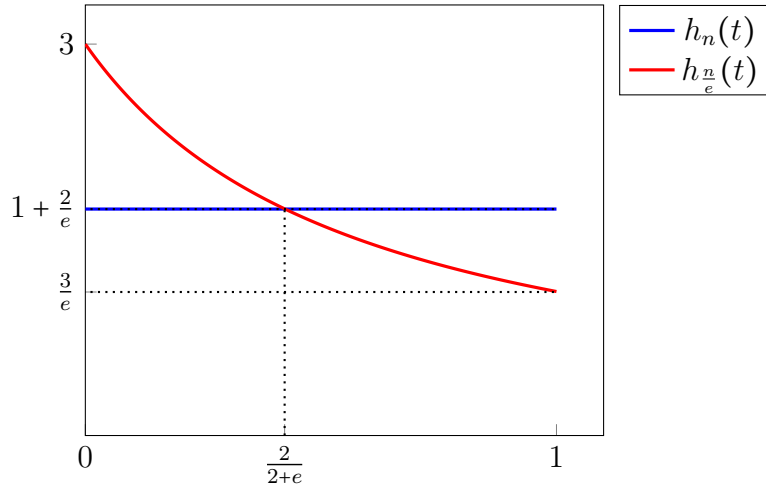


Figure 17: Hardness curves obtained from different values of  $t$ .

Using trivial, arithmetical operations, we calculate that those two curves intersect when  $t = \frac{2}{2+e}$ . Figure 18 shows the hardness of  $\ell$ -CENTRUM (and effectively ORDERED  $k$ -MEDIAN) we eventually prove.

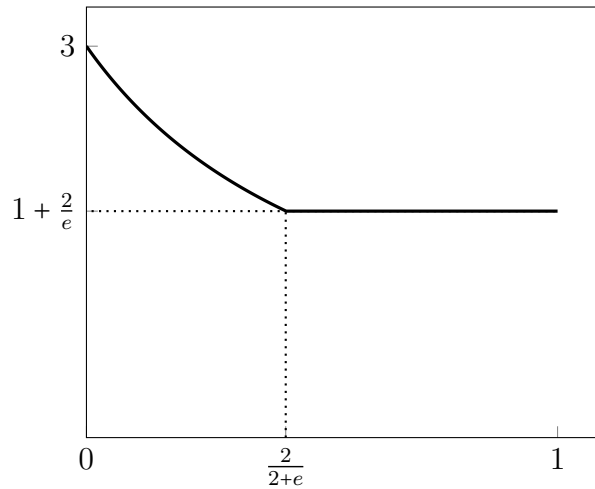


Figure 18: Proved hardness of the  $k$ -CENTDIAN and ORDERED  $k$ -MEDIAN problems approximation

Therefore, we prove the expected theorem that, unless  $\mathbf{P} = \mathbf{NP}$ , this is not possible to find the algorithm yielding  $h(t) - \epsilon$  approximation ratio for the ORDERED  $k$ -MEDIAN problem restricted by  $t$ , where  $h(t)$  is defined in the following way:

$$h(t) = \begin{cases} \frac{3}{1+(e-1)t} & \text{if } t < \frac{2}{2+e} \\ 1 + \frac{2}{e} & \text{if } t \geq \frac{2}{2+e} \end{cases}$$

□

Similarly, those results are not shown in any way to be tight.

## 6 Conclusions

As the last part of the thesis, we present the summary of our results and promising areas worth exploring in future research.

### 6.1 Summary

The work presented in this thesis connects two important research branches: fixed parameterized complexity and generalizing clustering problems. The main contributions focus on understanding the ORDERED  $k$ -MEDIAN concerning the ratio  $t$  between the smallest and the greatest element of the weights vector. The novel outcome of this thesis is:

1. Design of the coresets construction for the ORDERED  $k$ -MEDIAN problem. The algorithm, while inspired by [5, 47, 58], additionally considered more general metrics while remaining in the Euclidean spaces. Further, the construction is parameterized by the value of  $t$ .
2. Fixed-parameter trackable algorithm for the ORDERED  $k$ -MEDIAN problem. Although we maintain a superpolynomial dependency on  $k$  and  $d$ , and due to the limitation of the coresets construction, we stay to the Euclidean space, we managed to obtain a significant improvement in the approximation ratio. Additionally, we expect the algorithm to be practical because of its similarity with the experimental results from [5] and [13].
3. The hardness of the approximation ratio for the ORDERED  $k$ -MEDIAN (parametrized by  $t$ ) and the  $\ell$ -CENTRUM problems. The lower bound for the approximation ratio was obtained in the general metric space and even our results, restricted to the specific metrics, are far from matching this bound (except for  $\ell = 1$  and  $\ell = n$ ). However, the noticeable improvement is  $1 + \frac{2}{\epsilon} - \epsilon$  hardness bound for  $k$ -MEDIAN problem. That matches the bound from [27], but does not rely on the *Gap-ETH* assumption.

Figure 19 combines the obtained hardness and the approximation ratio. However, as mentioned earlier, the Euclidean variant (which is one analyzed in the coresets construction) might have a significantly better approximations ratio available. As an example, proved hardness of Euclidean  $k$ -MEDIAN is considerably lower [85].

### 6.2 Further work

Although this thesis introduces new concepts and increases the understanding of the ORDERED  $k$ -MEDIAN problem, we need to admit the limitations we met. We consider them as opportunities for further research. The most prominent directions are:

1. Designing algorithm for more effective approximately submodular maximization subject to a matroid constraint. This is still a new area, and only a few researchers decided to put effort into this approach [68, 69, 76]. Ideally, we

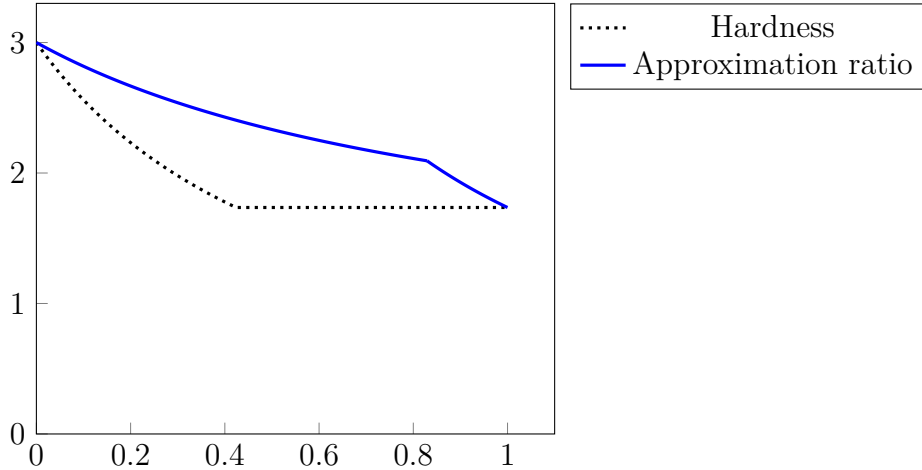


Figure 19: Proved hardness and approximation of the ORDERED  $k$ -MEDIAN problem with respect to the parameter  $t$ .

expect to find results matching the approximation ratio the general case (without the matroid constraint) known from [62]. Furthermore, we might try to take advantage of the specific properties of the matroid considered in our deductions. Namely, we do not consider a general matroid but a *partition matroid* [86], which may introduce better approximation techniques.

2. Improving coreset construction. Decreasing the size of the coreset may be possible. However, we should not target removing exponential dependency on  $d$  in the size of coreset due to the hardness bound discussed in the appendix of [4] (and then analyzed in [5]). The size of the coreset needs to be  $\Omega(\frac{k}{\epsilon^d})$ . Unfortunately, the hardness additionally impacts the general metric due to the general metric embedding with  $1 + \epsilon$  distortion into  $\mathbf{L}_\infty$  with unrestricted dimensionality (because of the well-known construction by Fréchet [87]).
3. Implementation and experiments of analyzed algorithms. Submodular maximization performed on the real-world data set yields surprisingly good results. We expect that the practical results may significantly overperform our proved bound regarding running time and approximation ratio.
4. Exploring unbounded heuristics. Inspired by our previous results for a related problem [13], we expect that unbounded methods (e.g., those employing local search or linear programming techniques with randomized rounding) may produce significantly better solutions (with a better running time) than those with bounded approximation ratio.
5. Exploring other FPT approaches. Although the framework constructed by [15] may be insufficient, the fixed-parameter tractable algorithms for clustering problems still constitute a new approach to clustering problems.
6. Formulating more generalized clustering problems. E.g., this is worth noticing the  $k$ -MEANS is not a particular case of ORDERED  $k$ -MEDIAN. The open

question is to formulate a natural generalization, which will include the  $k$ -MEANS problem.

7. Inspired by [88] and the application of ANTI-CENT-DIAN problem, we may consider exploring the ANTI-ORDERED  $k$ -MEDIAN problem, where the most remote client connects with the lowest weight. We expect that a similar framework used in the thesis may apply. However, the corests construction needs adjustments, because this objective is not a convex combination of  $\ell$ -CENTRUM instances.

## References

- [1] D. Chakrabarty and C. Swamy, “Approximation algorithms for minimum norm and ordered optimization problems,” *STOC 2019*, (New York, NY, USA), p. 126–137, Association for Computing Machinery, 2019.
- [2] J. Byrka, K. Sornat, and J. Spoerhase, “Constant-factor approximation for ordered k-median,” pp. 620–631, 11 2017.
- [3] V. Cohen-Addad, K. G. Larsen, D. Saulpic, and C. 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 2022*, (New York, NY, USA), p. 1038–1051, Association for Computing Machinery, 2022.
- [4] D. N. Baker, V. Braverman, L. Huang, S. H. Jiang, R. Krauthgamer, and X. Wu, “Coresets for clustering in graphs of bounded treewidth,” in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, vol. 119 of *Proceedings of Machine Learning Research*, pp. 569–579, PMLR, 2020.
- [5] V. Braverman, S. H. Jiang, R. Krauthgamer, and X. Wu, “Coresets for ordered weighted clustering,” *CoRR*, vol. abs/1903.04351, 2019.
- [6] K. Jain and V. V. Vazirani, “Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation,” *J. ACM*, vol. 48, p. 274–296, mar 2001.
- [7] A. Dehghan-Kooshkghazi, B. Kaminski, L. Krainski, P. Pralat, and F. Th  berge, “Evaluating node embeddings of complex networks,” *CoRR*, vol. abs/2102.08275, 2021.
- [8] J. Kleinberg, S. Mullainathan, and M. Raghavan, “Inherent trade-offs in the fair determination of risk scores,” 09 2016.
- [9] S. Bera, D. Chakrabarty, N. Flores, and M. Negahbani, “Fair algorithms for clustering,” in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alch  -Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.
- [10] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii, “Fair clustering through fairlets,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, *NIPS'17*, p. 5036–5044, 2017.
- [11] S. Thejaswi, B. Ordozgoiti, and A. Gionis, “Diversity-aware k-median : Clustering with fair center representation,” *CoRR*, vol. abs/2106.11696, 2021.
- [12] A. Gadekar, B. Ordozgoiti, and S. Thejaswi, “Approximation algorithms for k-median with lower-bound constraints,” *CoRR*, vol. abs/2112.07030, 2021.

- [13] S. Thejaswi, A. Gadekar, B. Ordozgoiti, and M. 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, (New York, NY, USA), p. 1749–1759, Association for Computing Machinery, 2022.
- [14] A. Chhabra, K. Masalkovaitė, and P. Mohapatra, “An overview of fairness in clustering,” *IEEE Access*, vol. 9, pp. 130698–130720, 2021.
- [15] V. Cohen-Addad, A. Gupta, A. Kumar, E. Lee, and J. Li, “Tight FPT approximations for  $k$ -median and  $k$ -means,” *CoRR*, vol. abs/1904.12334, 2019.
- [16] H. Ding and J. Xu, “A unified framework for clustering constrained data without locality property,” *Algorithmica*, vol. 82, 04 2020.
- [17] J. Cheetham, F. Dehne, A. Rau-Chaplin, U. Stege, and P. Taillon, “A parallel fpt application for clusters.,” pp. 70–77, 01 2003.
- [18] S. Bandyapadhyay, F. Fomin, P. A. Golovach, N. Purohit, and K. Simonov, “Fpt approximation for fair minimum-load clustering,” in *International Symposium on Parameterized and Exact Computation*, 2021.
- [19] A. Aouad and D. Segev, “The ordered  $k$ -median problem: Surrogate models and approximation algorithms,” *Math. Program.*, vol. 177, p. 55–83, sep 2019.
- [20] J. Puerto and A. Tamir, “Locating tree-shaped facilities using the ordered median objective,” *Math. Program.*, vol. 102, pp. 313–338, 03 2005.
- [21] V. Verter and A. Murat, “S. nickel and j. puerto: Location theory: a unified approach,” *Mathematical Methods of Operations Research*, vol. 66, no. 2, pp. 369–371, 2007.
- [22] J. Kalcsics, S. Nickel, J. Puerto, and A. Tamir, “Algorithmic results for ordered median problems,” *Oper. Res. Lett.*, vol. 30, pp. 149–158, 06 2002.
- [23] J. Kalcsics, S. Nickel, and J. Puerto, “Multifacility ordered median problems on networks: A further analysis,” *Networks*, vol. 41, pp. 1–12, 01 2003.
- [24] A. Tamir, “The  $k$ -centrum multi-facility location problem,” *Discrete Applied Mathematics*, vol. 109, no. 3, pp. 293–307, 2001.
- [25] J. Puerto and A. M. Rodríguez-Chía, *Ordered Median Location Problems*, pp. 261–302. Cham: Springer International Publishing, 2019.
- [26] M. Dyer and A. Frieze, “A simple heuristic for the  $p$ -center problem,” *Operations Research Letters*, vol. 3, pp. 285–288, 02 1985.
- [27] S. Guha and S. Khuller, “Greedy strikes back: Improved facility location algorithms,” *Journal of Algorithms*, vol. 31, no. 1, pp. 228–248, 1999.
- [28] M. R. Ackermann, J. Blömer, and C. Sohler, “Clustering for metric and nonmetric distance measures,” *ACM Trans. Algorithms*, vol. 6, sep 2010.

- [29] J. Byrka, T. Pensyl, B. Rybicki, A. Srinivasan, and K. Trinh, “An improved approximation for k-median and positive correlation in budgeted optimization,” vol. 13, mar 2017.
- [30] V. Cohen-Addad, H. Esfandiari, V. Mirrokni, and S. Narayanan, “Improved approximations for euclidean k-means and k-median, via nested quasi-independent sets,” in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, (New York, NY, USA), p. 1621–1628, Association for Computing Machinery, 2022.
- [31] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” 1967.
- [32] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” 1967.
- [33] I. S. Dhillon, Y. Guan, and B. Kulis, “Kernel k-means: Spectral clustering and normalized cuts,” in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, (New York, NY, USA), p. 551–556, Association for Computing Machinery, 2004.
- [34] S. Ahmadian, A. Norouzi-Fard, O. Svensson, and J. Ward, “Better guarantees for k-means and euclidean k-median by primal-dual algorithms,” *SIAM Journal on Computing*, vol. 49, no. 4, pp. FOCS17–97–FOCS17–156, 2020.
- [35] A. Kumar, Y. Sabharwal, and S. Sen, “A simple linear time  $(1+\epsilon)$ -approximation algorithm for k-means clustering in any dimensions,” in *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pp. 454–462, 01 2004.
- [36] D. Chakrabarty and C. Swamy, “Approximation algorithms for minimum norm and ordered optimization problems,” in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, (New York, NY, USA), p. 126–137, Association for Computing Machinery, 2019.
- [37] E. Minieka, “The m-center problem,” *SIAM Review*, vol. 12, no. 1, pp. 138–139, 1970.
- [38] O. Kariv and S. L. Hakimi, “An algorithmic approach to network location problems. i: The p-centers,” *SIAM Journal on Applied Mathematics*, vol. 37, no. 3, pp. 513–538, 1979.
- [39] J. N. Hooker, R. S. Garfinkel, and C. K. Chen, “Finite dominating sets for network location problems,” *Operations Research*, vol. 39, no. 1, pp. 100–118, 1991.
- [40] Z. Drezner and H. W. Hamacher, “Facility location - applications and theory,” Springer, 2001.
- [41] S. Li, *Approximating Capacitated k-median with  $(1 + \epsilon)k$  open facilities*, pp. 786–796.

- [42] M. Ghadiri, S. Samadi, and S. Vempala, “Socially fair k-means clustering,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT ’21, (New York, NY, USA), p. 438–448, Association for Computing Machinery, 2021.
- [43] M. Hajiaghayi, R. Khandekar, and G. Kortsarz, “Budgeted red-blue median and its generalizations,” in *Algorithms – ESA 2010* (M. de Berg and U. Meyer, eds.), (Berlin, Heidelberg), pp. 314–325, Springer Berlin Heidelberg, 2010.
- [44] R. Krishnaswamy, A. Kumar, V. Nagarajan, Y. Sabharwal, and B. Saha, “The matroid median problem,” SODA ’11, (USA), p. 1117–1130, Society for Industrial and Applied Mathematics, 2011.
- [45] S. A. Esmaeili, S. Duppala, J. P. Dickerson, and B. Brubach, “Fair labeled clustering,” in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD ’22, (New York, NY, USA), p. 327–335, Association for Computing Machinery, 2022.
- [46] J. Halpern, “The location of a centroidal convex combination on a undirected tree,” *Journal of Regional Science*, vol. 16, pp. 237–245, 1976.
- [47] S. Har-Peled and S. Mazumdar, “Coresets for  $k$ -means and  $k$ -median clustering and their applications,” *CoRR*, vol. abs/1810.12826, 2018.
- [48] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan, “Approximating extent measures of points,” *J. ACM*, vol. 51, p. 606–635, jul 2004.
- [49] P. K. Agarwal, C. M. Procopiuc, and K. R. Varadarajan, “Approximation algorithms for  $k$ -line center,” in *Algorithms — ESA 2002* (R. Möhring and R. Raman, eds.), (Berlin, Heidelberg), pp. 54–63, Springer Berlin Heidelberg, 2002.
- [50] M. Bundeinedoiu, S. Har-Peled, and P. Indyk, “Approximate clustering via core-sets,” in *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC ’02, (New York, NY, USA), p. 250–257, Association for Computing Machinery, 2002.
- [51] S. Har-Peled and K. Varadarajan, “Projective clustering in high dimensions using core-sets,” in *Proceedings of the Eighteenth Annual Symposium on Computational Geometry*, SCG ’02, (New York, NY, USA), p. 312–318, Association for Computing Machinery, 2002.
- [52] K. Chen, “On coresets for  $k$ -median and  $k$ -means clustering in metric and euclidean spaces and their applications,” *SIAM Journal on Computing*, vol. 39, no. 3, pp. 923–947, 2009.
- [53] C. Sohler and D. P. Woodruff, “Strong coresets for  $k$ -median and subspace approximation: Goodbye dimension,” *CoRR*, vol. abs/1809.02961, 2018.

- [54] O. Bachem, M. Lucic, and S. Lattanzi, “One-shot coresets: The case of k-clustering,” in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics* (A. Storkey and F. Perez-Cruz, eds.), vol. 84 of *Proceedings of Machine Learning Research*, pp. 784–792, PMLR, 09–11 Apr 2018.
- [55] D. Feldman, M. Monemizadeh, and C. Sohler, “A ptas for k-means clustering based on weak coresets,” in *Proceedings of the Twenty-Third Annual Symposium on Computational Geometry*, SCG ’07, (New York, NY, USA), p. 11–18, Association for Computing Machinery, 2007.
- [56] D. Feldman and T. Tassa, “More constraints, smaller coresets: Constrained matrix approximation of sparse big data,” in *KDD 2015 - Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 249–258, Association for Computing Machinery, Aug. 2015. Publisher Copyright: © 2015 ACM.; 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2015 ; Conference date: 10-08-2015 Through 13-08-2015.
- [57] X. Wu, “Efficient Implementation of Coreset-based K-Means Methods,” master’s thesis, Aalto University. School of Science, 2021.
- [58] S. Har-Peled and A. Kushal, “Smaller coresets for k-median and k-means clustering,” *Discrete & Computational Geometry*, vol. 37, pp. 3–19, 12 2007.
- [59] J. Matousek, *Lectures on Discrete Geometry*. Graduate Texts in Mathematics, Springer New York, 2013.
- [60] Y. Wu, “Information-theoretic methods in high-dimensional statistics, packing, covering, and consequences on minimax risk.” <http://www.stat.yale.edu/~yw562/teaching/598/lec14.pdf>, 2006. Accessed: 2022–10-03.
- [61] D. Feldman and M. Langberg, “A unified framework for approximating and clustering data,” *CoRR*, vol. abs/1106.1379, 2011.
- [62] A. Das and D. Kempe, “Approximate submodularity and its applications: Subset selection, sparse approximation and dictionary selection,” *Journal of Machine Learning Research*, vol. 19, no. 3, pp. 1–34, 2018.
- [63] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, vol. B. 01 2003.
- [64] T. Horel and Y. Singer, “Maximization of approximately submodular functions,” in *Advances in Neural Information Processing Systems* (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.

- [65] Y. Singer and A. Hassidim, “Optimization for approximate submodularity,” in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.
- [66] N. Veldt, A. R. Benson, and J. Kleinberg, “Approximate decomposable submodular function minimization for cardinality-based components,” 2021.
- [67] F. Chierichetti, A. Dasgupta, and R. Kumar, “On additive approximate submodularity,” 2020.
- [68] Q. Nong, T. Sun, S. Gong, Q. Fang, D. Du, and X. Shao, “Maximize a monotone function with a generic submodularity ratio,” in *Algorithmic Aspects in Information and Management* (D.-Z. Du, L. Li, X. Sun, and J. Zhang, eds.), (Cham), pp. 249–260, Springer International Publishing, 2019.
- [69] S. Gong, Q. Nong, W. Liu, and Q. Fang, “Parametric monotone function maximization with matroid constraints,” *Journal of Global Optimization*, vol. 75, 11 2019.
- [70] J. EDMONDS, “Submodular functions, matroids, and certain polyhedra,” *Combinatorial Structures and Their Applications*, 1970.
- [71] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, *An analysis of approximations for maximizing submodular set functions—II*, pp. 73–87. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978.
- [72] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, “Maximizing a monotone submodular function subject to a matroid constraint,” *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1740–1766, 2011.
- [73] G. L. Nemhauser and L. A. Wolsey, “Best algorithms for approximating the maximum of a submodular set function,” *Mathematics of Operations Research*, vol. 3, no. 3, pp. 177–188, 1978.
- [74] Y. Filmus and J. Ward, “A tight combinatorial algorithm for submodular maximization subject to a matroid constraint,” in *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pp. 659–668, 2012.
- [75] A. Ene and H. L. Nguyen, “Towards nearly-linear time algorithms for submodular maximization with a matroid constraint,” in *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece* (C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, eds.), vol. 132 of *LIPICs*, pp. 54:1–54:14, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- [76] B. Liu and M. Hu, “Fast algorithms for maximizing monotone nonsubmodular functions,” *Journal of Combinatorial Optimization*, vol. 43, 07 2022.
- [77] L. Nguyen and M. Thai, “Efficient algorithms for monotone non-submodular maximization with partition matroid constraint,” pp. 4774–4780, 07 2022.

- [78] G. Nemhauser, L. Wolsey, and M. Fisher, “An analysis of approximations for maximizing submodular set functions—i,” *Mathematical Programming*, vol. 14, pp. 265–294, 12 1978.
- [79] A. Kumar, “Capacitated k-center problem with vertex weights,” in *FSTTCS*, 2016.
- [80] N. Megiddo and A. Tamir, “New results on the complexity of p-center problems,” *SIAM J. Comput.*, vol. 12, pp. 751–758, 1983.
- [81] A. Bhattacharya, D. Goyal, and R. Jaiswal, “Hardness of approximation of euclidean k-median,” *CoRR*, vol. abs/2011.04221, 2020.
- [82] N. Megiddo and K. J. Supowit, “On the complexity of some common geometric location problems,” *SIAM Journal on Computing*, vol. 13, no. 1, pp. 182–196, 1984.
- [83] R. M. Karp, *Reducibility among Combinatorial Problems*, pp. 85–103. Boston, MA: Springer US, 1972.
- [84] Feige, “A threshold of  $\ln n$  for approximating set cover,” *J. ACM*, vol. 45, p. 634–652, jul 1998.
- [85] A. Bhattacharya, D. Goyal, and R. Jaiswal, “Hardness of Approximation for Euclidean k-Median,” in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)* (M. Wooters and L. Sanità, eds.), vol. 207 of *Leibniz International Proceedings in Informatics (LIPIcs)*, (Dagstuhl, Germany), pp. 4:1–4:23, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.
- [86] A. Recski, “Maps of matroids with applications,” *Discrete Mathematics*, vol. 303, no. 1, pp. 175–185, 2005. The 2002 Korea-Hungary Joint Workshop on Combinatorics and The 2002 Com2MaC Conference on Graphs and Combinatorics.
- [87] M. Fréchet, “Les dimensions d’un ensemble abstrait,” *Mathematische Annalen*, vol. 68, pp. 145–168, 1910.
- [88] M. Colebrook and J. Sicilia, “An  $o(mn)$  algorithm for the anti-cent-dian problem,” *Applied Mathematics and Computation*, vol. 183, pp. 350–364, 12 2006.