Near Linear Time Approximation Schemes for Clustering of Partially Doubling Metrics

Anne Driemel
University of Bonn, Germany

Jan Höckendorff
University of Cologne, Germany

Ioannis Psarros
Archimedes, Athena Research Center, Greece

Christian Sohler
University of Cologne, Germany

Di Yue
University of Toronto, Canada

Abstract

In the metric k-median problem we are given a finite metric space $(X \cup Y, \mathbf{d})$ and the objective is to compute a set of k centers $C \subseteq Y$ that minimizes $\sum_{p \in X} \min_{c \in C} \mathbf{d}(p, c)$. In general metric spaces, the best polynomial time algorithm, which is due to Cohen-Addad, Grandoni, Lee, Schwiegelshohn, and Svensson [17], computes a $(2+\varepsilon)$-approximation for arbitrary constant $\varepsilon > 0$. However, if the metric space has bounded doubling dimension, a near linear time $(1+\varepsilon)$-approximation algorithm is known due to the work of Cohen-Addad, Feldmann, and Saulpic [16].

In this paper, we show that the $(1+\varepsilon)$-approximation algorithm can be generalized to the case when either X or Y has bounded doubling dimension (but the other set not). The case when X has bounded doubling dimension is motivated by the assumption that even though X is part of a high-dimensional space, it may be that it is close to a low-dimensional structure. The case when Y has bounded doubling dimension is perhaps more natural. It is motivated by specific clustering problems where the centers are low-dimensional. Specifically, our work in this setting implies the first near linear time approximation algorithm for the (k, ℓ)-median problem under discrete Fréchet distance when ℓ is constant. The latter problem is a version of the k-median problem under Fréchet distance when the input consists of time series of z reals and where the centers are time series of ℓ reals [21]. Previously, for this problem no $(1+\varepsilon)$-approximation algorithm with running time polynomial in k was known. We also introduce a novel complexity reduction for time series of real values that leads to a similar result for the case of discrete Fréchet distance.

In order to solve the case when Y has a bounded doubling dimension, we introduce a form of dimension reduction that replaces points from X by sets of points in Y. To solve the case when X has a bounded doubling dimension, we generalize Talwar's decomposition [41] of doubling metrics to our setting. The running time of our algorithms is $2^{2^t}\tilde{O}(n+m)$ where $t = O(\mathrm{ddim} \log \frac{\mathrm{ddim}}{\varepsilon})$ and where ddim is the doubling dimension of X (resp. Y). The results also extend to the metric (uncapacitated) facility location problem. We believe that our techniques are likely applicable to other problems.

2012 ACM Subject Classification Theory of computation → Facility location and clustering

Keywords and phrases Approximation Algorithms, Doubling Spaces, Facility Location, k-Median, Discrete Fréchet Distance

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.80

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2603.24336

53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026).
Editors: Sayan Bhattacharya, Danupon Nanongkai, Michael Benedikt, and Gabriele Puppis;
Article No. 80; pp. 80:1–80:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Funding *Anne Driemel*: Affiliated with Lamarr Institute for Machine Learning and Artificial Intelligence; Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project Number 459420781.
Jan Höckendorff: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project Number 459420781.
Ioannis Psarros: This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.
Christian Sohler: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project Number 459420781.
Di Yue: Funded by key profile area Intelligent Methods for Earth System Science at University of Cologne.

1 Introduction

Partitioning of data sets according to data characteristics is one of the most fundamental problems in data analysis and optimization. Depending on the underlying problem there are many different variants of partitioning problems. In this paper, we consider the metric k-median problem and the (closely related) metric facility location problem. The former problem belongs to the area of unsupervised learning and is formulated as follows. We are given a metric space $(X \cup Y, \mathbf{d})$, where the set X consists of the data points and the set Y of allowed center locations. The goal is to choose a set C of k centers from Y such that the $\text{cost}_k(X, C) = \sum_{x \in X} \min_{c \in C} \mathbf{d}(x, c)$ is minimized. The set C induces a partition of X into k sets by assigning each point to its nearest center. The resulting partitioning of X is also called a *clustering*. The facility location problem is closely related even though it originates from a different setting. In facility location we have a set of clients X and a set of possible facilities Y both from a metric space $(X \cup Y, \mathbf{d})$ and with each $f \in Y$ there is an opening cost $\text{ocost}(f)$ associated. Every client has to be assigned to an open facility and will pay the distance to the facility as connection cost. The objective is to find a set of facilities $F \subset Y$ such that the sum of opening and connection costs is minimized, that is, we want to minimize $\sum_{x \in X} \min_{f \in F} \mathbf{d}(x, f) + \sum_{f \in F} \text{ocost}(f)$. Thus, the main difference between the two problems is, that in k-median clustering the number of centers is restricted to k, while in facility location we may open as many centers as we want, but we need to pay for it. Otherwise, both objective functions minimize the sum of distances of points from X to their nearest centers.

Neither problem admits a polynomial time $(1 + \varepsilon)$-approximation algorithm for arbitrary small constant $\varepsilon > 0$ under standard complexity theoretical assumptions. In fact, the k-median problem cannot be approximated better than $1 + 2/e$ [32] and the facility location problem not better than 1.463 [26, 32]. At the same time, both problems can be fairly well approximated. For the k-median problem the best possible approximation algorithm achieves a factor $2 + \varepsilon$ approximation [17] and for the facility location problem, the best known approximation factor is 1.488 [35]. Interestingly, both problems can be approximated up to a factor of $(1 + \varepsilon)$, if the underlying metric space has bounded doubling dimension [16].

In this paper, we raise the question whether a $(1 + \varepsilon)$-approximation can also be achieved, if only one of the sets has bounded doubling dimension, but the other set is high-dimensional. This setting has been studied before in the context of nearest neighbor search [31, 28] and Euclidean facility location [30]. In this paper, we give the first near-linear time $(1 + \varepsilon)$-approximation algorithms for facility location and k-median in these settings.

We then observe that other known clustering problems have centers coming from a space that is low-dimensional: An example is the (k, ℓ)-median clustering problem [21] of time series data under the discrete Fréchet distance. The Fréchet distance is a standard distance measure for polygonal curves. These curves can be represented as a time series of points. The Fréchet distance is especially suited for comparing series of different complexity (number of points in the time series).

In the (k, ℓ)-median problem the length of the center time series is restricted to a constant ℓ, which essentially means that the space of center time series is doubling (albeit it does not imply that the space of clients is doubling).

We then apply our algorithm to get the first $(1 + \varepsilon)$-approximation algorithm for this problem with a near linear running time (for constant ℓ) while all previous algorithms were exponential in k [6, 39]. For the case of one-dimensional ambient space we also give a more direct algorithm that is based on a new complexity reduction method for the discrete Fréchet distance that reduces the problem to the case of bounded doubling dimension. We further investigate the reverse setting when the data points are low-dimensional and the candidate center set is high dimensional. Such a setting may arise when high-dimensional data is located on or near a low-dimensional structure, which is a common assumption, for example, in the field of manifold learning. We show that also in the reverse setting we obtain an almost linear time approximation scheme.

This paper is a direct follow up to the work of Driemel et al. [20] that introduced a complexity reduction for time series and presented a near linear time $(1 + \varepsilon)$-approximation algorithm for (k, ℓ)-median for time series. This paper strictly improves both results by giving an explicit bound to the complexity reduction and generalizing the $(1 + \varepsilon)$-approximation to polygonal curves of arbitrary ambient dimension.

2 Problem Definitions

In this section we introduce the problems we consider. We start by defining the metric facility location problem. Let $(X \cup Y, \mathbf{d})$ be a metric space. Input to the facility location problem is a set of n data points (or clients) X and a set of m candidate facilities Y. Every facility $f \in Y$ is assigned with an opening cost $\mathrm{ocost}(f) > 0$. The goal is to output a set of facilities $F \subseteq Y$, such that

$$\mathrm{cost}_{\mathrm{fl}}(X, F) := \sum_{x \in X} \mathbf{d}(x, F) + \sum_{f \in F} \mathrm{ocost}(f)$$

is minimized, where $\mathbf{d}(x, F) := \min_{f \in F} \mathbf{d}(x, f)$. The optimal facility location value is denoted by $\mathrm{opt}_{\mathrm{fl}}(X, Y) := \min_{F \subseteq Y} \mathrm{cost}_{\mathrm{fl}}(X, F)$.

In the metric k-median problem, our input is also a set of n data points X and a set of m candidate centers Y. The objective is to find a subset $F \subseteq Y$ of centers such that

$$\mathrm{cost}_k(X, F) := \sum_{x \in X} \mathbf{d}(x, F)$$

is minimized over all sets $F \subseteq Y$ of cardinality k. The optimal k-median value is denoted by $\mathrm{opt}_k(X, Y) := \min_{F \subseteq Y, |F| \le k} \mathrm{cost}_k(X, F)$.

Throughout this paper, unless stated otherwise, we assume that the distances in $X \cup Y$ can be computed in constant time. We remark that in a setting where this is not the case and we require time T to compute a distance between two points in $X \cup Y$ then we get an additional factor of T in the running time. It is known that if the metric space $(X \cup Y, \mathbf{d})$

has bounded doubling dimension, we have a near linear time approximation algorithm [16]. In this paper we study the problem variant, where just one of the two sets X and Y has bounded doubling dimension. Perhaps surprisingly, we show that in these cases we can still obtain a $(1+\varepsilon)$-approximation algorithm. Since one can $O(1)$-approximate the doubling dimension of a metric in $2^{O(\mathrm{ddim})} n \log n$ time [29], we will assume that our algorithms are given an upper bound ddim on the doubling dimension.

Application: The (k,ℓ)-Median Problem under Discrete Fréchet Distance

We would like to apply our results to a geometric variant of k-median clustering that we will define in the following. We denote a polygonal curve π with vertices $p_1,\dots,p_z$ in $\mathbb{R}^d$ as $\pi = \langle p_1, p_2, \dots, p_z\rangle$. The *complexity* of a polygonal curve is the number of points of the sequence defining it. The set of all polygonal curves of complexity z with vertices in $\mathbb{R}^d$ is denoted by $\mathbb{X}_z^d$. A *traversal* T between a polygonal curve of complexity z and a polygonal curve of complexity ℓ is a sequence of index pairs $T = (i_1, j_1), (i_2, j_2), \dots, (i_t, j_t)$ such that the following conditions hold:

1. $(i_1, j_1) = (1, 1)$,
2. $(i_t, j_t) = (z, \ell)$, and for each $1 \le r < t$,
3. $i_{r+1} - i_r \in \{0, 1\}$,
4. $j_{r+1} - j_r \in \{0, 1\}$,
5. $(i_{r+1} - i_r) + (j_{r+1} - j_r) \ge 1$.

Let $\mathcal{T}_{z,\ell}$ be the set of all traversals between polygonal curves of complexity z and polygonal curves of complexity ℓ.

▶ **Definition 1.** *The* discrete Fréchet distance *between* $\pi = \langle p_1,\dots,p_z\rangle$ *and* $\tau = \langle q_1,\dots,q_\ell\rangle$ *is defined as:* $\mathbf{d}_{dF}(\pi,\tau) = \min_{T\in\mathcal{T}_{z,\ell}} \max_{(i,j)\in T} \|p_i - q_j\|_2$.

We consider the (k,ℓ)-median problem for clustering under the discrete Fréchet distance that has been introduced in [21] in the context of the continuous Fréchet distance. The problem is a variant of the k-median problem under Fréchet distance, where the complexity of the center polygonal curves is restricted to be at most ℓ.

▶ **Definition 2** ((k,ℓ)-median clustering problem)**.** *Given a set of polygonal curves* $\Pi \subset \mathbb{X}_z^d$ *and parameters* $k,\ell \in \mathbb{N}$*, compute a set* $\mathcal{C} \subset \mathbb{X}_\ell^d$*,* $|\mathcal{C}| = k$*, that minimizes* $\sum_{\pi\in\Pi} \min_{\tau\in\mathcal{C}} \mathbf{d}_{dF}(\pi,\tau)$.

We remark that one may invariantly define the centers to have complexity at most ℓ, instead of exactly ℓ. This does not change the problem, since every polygonal curve with complexity fewer than ℓ can be extended to a polygonal curve with ℓ vertices by repeating the first element of the polygonal curve without affecting the discrete Fréchet distance.

3 Our Results

In the following section, we present our results. We first show that for the k-median and the facility location problem there are near-linear time $(1+\varepsilon)$-approximation algorithms when the set of facilities Y has bounded doubling dimension. This extends previous results that were restricted to the case that $X \cup Y$ has bounded doubling dimension. The Theorem below summarizes [19, Theorems 5.1 and 7.1] and is proven in the corresponding sections of the full version [19].

▶ **Theorem 3.** *There are randomized algorithms that, given as input* $\varepsilon \in (0, \frac{1}{2})$*,* $n, m, k \in \mathbb{N}$ *and* $(X \cup Y, \mathbf{d})$ *with* $|X| = n, |Y| = m, \mathrm{ddim}(Y) \le \mathrm{ddim}$*, compute a* $(1+\varepsilon)$*-approximation* *of the k-median and facility location problem in time* $2^{2^t} \cdot \tilde{O}(n+m)$ *with constant success probability, where* $t \in O(\operatorname{ddim} \log(\operatorname{ddim}/\varepsilon))$.

We then show a similar result for the case when the set of clients X has bounded doubling dimension. We keep the theorems separated since the underlying techniques are different. The following Theorem summarizes [19, Theorems 6.1 and 8.1] and is proven in the corresponding sections of the full version [19].

▶ **Theorem 4.** *There are randomized algorithms that, given as input* $\varepsilon \in (0, \frac{1}{2})$, $n, m, k \in \mathbb{N}$ *and* $(X \cup Y, \mathbf{d})$ *with* $|X| = n, |Y| = m, \operatorname{ddim}(X) \leq \operatorname{ddim}$, *compute a* $(1+\varepsilon)$*-approximation of the k-median and facility location problem in time* $2^{2^t} \cdot \tilde{O}(n+m)$ *with constant success probability, where* $t \in O(\operatorname{ddim} \log(\operatorname{ddim}/\varepsilon))$.

We apply our result to the (k, ℓ)-median problem under discrete Fréchet distance that is the first near linear time $(1+\varepsilon)$-approximation algorithm for this problem when ℓ and d (the dimension of the ambient space) are constant. All prior algorithms were exponential in k.

▶ **Theorem 5.** *There is a randomized algorithm that, given as input* $\varepsilon \in (0, \frac{1}{2})$, $n, z, k, \ell \in \mathbb{N}$, $P \subset \mathbb{X}_z^d$ *with* $|P| = n$ *computes a* $(1+\varepsilon)$*-approximate solution to the* (k, ℓ)*-median problem in time* $2^{2^t} \cdot \tilde{O}(d\ell n z)$ *with constant success probability, where* $t \in O(d\ell \log(d\ell/\varepsilon))$.

4 Technical Overview

In the following we give an overview of the main ideas of our results. There are two conceptual ideas, one related to the case of low-dimensional centers, the other one related to low-dimensional clients. In the setting of low-dimensional centers, we develop a form of dimension reduction which represents every high-dimensional client by a set of points in the low-dimensional space. In the setting of low-dimensional clients, our main contribution is a new hierarchical decomposition that generalizes Talwar's decomposition [41] to partially doubling metric spaces. Both ideas are independent of the concrete problems we study and we therefore believe that there is high potential to apply them to other problems. Once we have established these main ideas, there is the technical challenge to integrate them into the dynamic programming approach from [16]. This requires overcoming several technical problems to deal with our setting.

We start by describing the case of low-dimensional centers. We illustrate our ideas on the facility location problem. For simplicity, we will assume uniform opening costs. The approach to k-median is similar. Finally, we discuss an application to the (k, ℓ)-median clustering problem under discrete Fréchet distance.

4.1 Low-dimensional Centers

We consider the case where the point set X is in an arbitrary metric space and Y has bounded doubling dimension ddim. A similar setting has been considered before in [25] where the authors develop approximation algorithms for finding the smallest set to be removed to get a set of bounded doubling dimension and where they show that one can compute an approximate minimum spanning tree and other proximity structures when $O(\sqrt{n})$ points are not doubling. A simple illustrative example is to have a point set X in $\mathbb{R}^d$ with Euclidean distance and a fixed 2-dimensional plane Y that is supposed to contain the centers. In this example, the set Y will be unbounded (one can usually discretize such a space to obtain a set of candidates that contains a $(1+\varepsilon)$-approximation).

Dimension reduction. A simple idea to solve this special case is to project all points from X to the plane Y and solve the resulting low-dimensional problem. Such an approach will result in a constant approximation as argued in the following. For the analysis we can think of the projection as moving all points to Y. By the triangle inequality, this will change the cost of any solution by at most the sum of distances the points have been moved. Since this sum is also a lower bound on the connection cost of any solution to the high-dimensional input, we get that any solution on the projection is a constant approximation. For similar reasons approximating the distances from x to $y \in Y$ by $\mathbf{d}(x, \pi_Y(x)) + \mathbf{d}(\pi_Y(x), y)$, where $\pi_Y(x)$ denotes the point from Y closest to x, only gives a constant approximation. Instead of replacing x by a single point, our new idea is to replace x by a set of points N_x (which we call *proxies*), which is an $\varepsilon\, \mathbf{d}(x, \pi_Y(x))$-net of a ball $B_Y(\pi_Y(x), \mathbf{d}(x, \pi_Y(x))/\varepsilon) \subseteq Y$ around $\pi_Y(x)$ of radius $\mathbf{d}(x, \pi_Y(x))/\varepsilon$. (Formally, a ρ-net of a set B is a subset $N \subseteq B$, such that 1) the interpoint distances in N are at least ρ, and 2) every point in B has a nearby point in N within distance ρ.) Taking $\min_{u \in N_x} \mathbf{d}(x, u) + \mathbf{d}(u, y)$ for $y \in Y$ we approximate all distances within a factor of $(1 + \varepsilon)$ as summarized in the following lemma.

▶ **Lemma 6** (Informal version of [19, Lemma 3.2]). *Let $\varepsilon \in (0, \frac{1}{2})$. For every $x \in X$ let N_x be an $\varepsilon \cdot \mathbf{d}(x, Y)$-net of the ball $B_Y(\pi_Y(x), \mathbf{d}(x, Y)/\varepsilon)$ with $\pi_Y(x) \in N_x$. For all $x \in X$ and $y \in Y$ define*

$$\widehat{\mathbf{d}}(x, y) = \min_{u \in N_x} \mathbf{d}(x, u) + \mathbf{d}(u, y). \tag{1}$$

Then we have

$$\mathbf{d}(x, y) \le \widehat{\mathbf{d}}(x, y) \le (1 + 4\varepsilon) \cdot \mathbf{d}(x, y).$$

For technical reasons, our algorithm does not directly compute the proxy set N_x around the exact nearest neighbor $\pi_Y(x)$ of x in Y. Instead, we will first compute an $O(1)$-approximate solution $S \subseteq Y$, find in S an $O(1)$-approximate nearest neighbor s of x, and construct N_x as an $\varepsilon\, \mathbf{d}(x, s)$-net of the ball $B_Y(s, \mathbf{d}(x, s)/\varepsilon)$. This replacement introduces an extra $\varepsilon\, \mathbf{d}(x, S)$ additive error in Lemma 6, which will not affect the correctness of our analysis. We note that conceptually similar ideas also appear in [16].

We combine the above dimension reduction idea with the algorithm of [16]. We start from a standard hierarchical decomposition for doubling metrics by Talwar [41], which is an analogue to the randomly-shifted quadtree in Euclidean space. The decomposition $\mathcal{H}$ is constructed on top of Y, and has $L = O(\log m)$ levels. For each $0 \le \ell \le L$, level ℓ of $\mathcal{H}$, denoted by $\mathcal{H}_\ell$, is a partition of Y into clusters of diameter at most 2^ℓ, and $\mathcal{H}_{\ell-1}$ is a refinement of $\mathcal{H}_\ell$. $\mathcal{H}$ can be represented as a tree, where each node corresponds to a cluster and has $2^{O(\mathrm{ddim})}$ children. In [16], a dynamic program is run on $\mathcal{H}$ to compute a $(1 + \varepsilon)$-approximation for facility location. In our setting, one immediate issue is that clients in X are not directly defined on $\mathcal{H}$, but are replaced by a proxy set $N_x \subseteq Y$. Therefore, it is not immediately clear that one can follow a similar approach as [16].

A brief review of [16]. For the sake of presentation, let us first have a brief review of the argument of [16], for the case $X = Y$. They use the notion of *portals*, which originates from [2]. Roughly speaking, the portal set P_C for a cluster $C \in \mathcal{H}_\ell$ is an $\varepsilon 2^\ell$-net of C. The level ℓ portals are the union of P_C over $C \in \mathcal{H}_\ell$. For a pair of points $x, y \in Y$, the actual distance between x and y will be replaced by the *portal-respecting distance* $\mathbf{d}^{\mathcal{H}}_{\mathrm{port}}(x, y)$, which is the length of the *portal-respecting path* between x and y. Specifically, let ℓ be the highest level where $\{x, y\}$ is *cut* w.r.t. $\mathcal{H}$, i.e., the highest level where x and y fall into different clusters. The portal-respecting path between x and y starts from x (a portal at level 0), each step connecting the current portal to the closest portal at one level up, until reaching a portal p at level ℓ. Portal p is then connected to another portal q at the same level, and the path goes all the way down from q to y. It can be shown that the portal-respecting distance $\mathbf{d}^{\mathcal{H}}_{\text{port}}(x, y)$ is upper bounded by $\mathbf{d}(x, y) + O(\varepsilon)2^{\ell}$. A number of techniques are developed in [16] to bound ℓ, the level where $\{x, y\}$ is *cut*.

Finally, a dynamic program is run on $\mathcal{H}$ w.r.t. portal-respecting distance $\mathbf{d}^{\mathcal{H}}_{\text{port}}$. Each entry of the DP table is encoded by a cluster C of $\mathcal{H}$, and a configuration $\mathbf{a}_C$ indicating how C interacts with other clusters as well as the current solution (facility set) F via its portals P_C. Roughly speaking, the configuration is a vector $\mathbf{a}_C \in \mathbb{R}^{P_C}$, whose p-th coordinate $\mathbf{a}_C(p)$ equals to the minimum portal-respecting distance between portal p and facilities in F. The value stored in table entry $(C, \mathbf{a}_C)$, denoted by $f(C, \mathbf{a}_C)$, is defined as the minimum facility location cost inside of C, given the configuration $\mathbf{a}_C$, i.e.,

$$f(C, \mathbf{a}_C) = \min_{\substack{F:\ F \text{ is consistent} \\ \text{with } \mathbf{a}_C}} \left\{ \sum_{x \in C} \mathbf{d}^{\mathcal{H}}_{\text{port}}(x, F) + \text{ocost}(F \cap C) \right\}.$$

To compute the DP table $f(C, \mathbf{a}_C)$, the algorithm enumerates all children D of C and all possible configurations. If the configurations $\{\mathbf{a}_D : D \text{ is a child of } C\}$ are consistent with $\mathbf{a}_C$, and the sum of $f(D, \mathbf{a}_D)$ over all children D is less than the current value of $f(C, \mathbf{a}_C)$, it then updates $f(C, \mathbf{a}_C)$ to be the sum of $f(D, \mathbf{a}_D)$.

Integrating the dimension reduction with portal-respecting distance. In our setting, if we would like to combine our approach with the algorithm of [16], the first step is to combine our dimension reduction idea with the portal-respecting distance. This is done in a rather straightforward way, by modifying (1) to

$$\widehat{\mathbf{d}}^{\mathcal{H}}_{\text{port}}(x, y) = \min_{u \in N_x} \mathbf{d}(x, u) + \mathbf{d}^{\mathcal{H}}_{\text{port}}(u, y).$$

Now, a main challenge is to bound the error incurred by the portal-respecting distance, namely, $\widehat{\mathbf{d}}^{\mathcal{H}}_{\text{port}}(x, y) - \mathbf{d}(x, y)$. In [16], this can be done by bounding the highest cutting level for a single pair of points $\{x, y\}$. However, since x is now represented by N_x, we have to bound the cutting level of $\{u, y\}$ for every $u \in N_x$. Nevertheless, we argue that the above error can still be effectively bounded, by only considering the cutting level for two sets $B_Y(\pi_Y(x), \mathbf{d}(x, Y)/\varepsilon)$ and $\{\pi_Y(x), y\}$. Specifically, if j is the highest level where $B_Y(\pi_Y(x), \mathbf{d}(x, Y)/\varepsilon)$ is cut, and ℓ is the highest level where $\{\pi_Y(x), y\}$ is cut, then we show that

$$\widehat{\mathbf{d}}^{\mathcal{H}}_{\text{port}}(x, y) - \mathbf{d}(x, y) \le O(\varepsilon)2^{\max\{j, \ell\}}.$$

Dynamic program. We further integrate the dimension reduction with the dynamic program framework in [16]. The algorithm is run on $\mathcal{H}$. Whenever we compute the connection cost from a client $x \in X$ to some set of facilities $F \subseteq Y$, we replace $\mathbf{d}(x, F)$ with $\widehat{\mathbf{d}}^{\mathcal{H}}_{\text{port}}(x, F)$. Two challenges come with such a replacement. First, recall that $\widehat{\mathbf{d}}^{\mathcal{H}}_{\text{port}}(x, F)$ depends on $\mathbf{d}^{\mathcal{H}}_{\text{port}}(u, F)$ for all proxies $u \in N_x$; therefore, to compute it we have to enumerate all $u \in N_x$, which can be done only when N_x is entirely contained in some cluster C. This is different from [16], where the connection cost of x can be trivially computed at the leaf node $\{x\}$. Second, even if we successfully define such a cluster C which contains N_x, it is unclear how we access the distance $\mathbf{d}^{\mathcal{H}}_{\text{port}}(u, F)$.

To resolve the first issue, we find for every $x \in X$ a suitable cluster $C(x)$ that entirely contains N_x in the preprocessing stage, and "defer" the computation of $\widehat{\mathbf{d}}^{\mathcal{H}}_{\mathrm{port}}(x, F)$ to cluster $C(x)$. More concretely, for a client $x \in X$ and a cluster C on $\mathcal{H}$, we say x is *revealed* in C if $B_Y(\pi_Y(x), \mathbf{d}(x, Y)/\varepsilon) \subseteq C$, and say x is *newly revealed* in C, denoted $C(x) = C$, if C is the lowest level cluster where x is revealed. Then each DP table entry $(C, \mathbf{a}_C)$ is defined to be the minimum *revealed facility location cost* of C, given the configuration $\mathbf{a}_C$, i.e.,

$$g(C, \mathbf{a}_C) := \min_{\substack{F:\, F \text{ is consistent} \\ \text{with } \mathbf{a}_C}} \left\{ \sum_{x \in X:\, x \text{ is revealed in } C} \widehat{\mathbf{d}}^{\mathcal{H}}_{\mathrm{port}}(x, F) + \mathrm{ocost}(F \cap C) \right\}.$$

Since all clients are revealed in the largest cluster (root node) Y, the root node stores the optimal facility location cost.

For the second issue, let us consider how the DP table is updated. The revealed cost of C can be decomposed into two parts – (1) the total revealed cost of C's child clusters, and (2) the *newly revealed cost* of C, i.e., $\sum_{x:\, C(x)=C} \widehat{\mathbf{d}}^{\mathcal{H}}_{\mathrm{port}}(x, F)$. The first part can be obtained from the corresponding table entries of C's children. For the second part, the key observation is that if x is newly revealed in C ($C(x) = C$), then the proxy set of x is entirely contained in the portal set of C, i.e., $N_x \subseteq P_C$. Therefore, for each $u \in N_x$, $\mathbf{d}^{\mathcal{H}}_{\mathrm{port}}(u, F)$ can be directly obtained from the configuration $\mathbf{a}_C$ which has the information of the connection of P_C, and thus $\widehat{\mathbf{d}}^{\mathcal{H}}_{\mathrm{port}}(x, F)$ can be effectively computed.

Our final algorithm has a running time of $2^{2^t} \tilde{O}(n + m)$ where $t = O(\mathrm{ddim} \log \frac{\mathrm{ddim}}{\varepsilon})$. We remark that this is faster than the algorithm of [16], which has t^2 instead of t in the (double) exponent. This improvement comes from the fact that one can improve the analysis of their hierarchical composition [16]. The fact, that such an improvement is possible had been observed in [10] referring to the paper [1]. We provide a self-contained proof of this fact in the full version [19, Lemma 2.5].

4.2 Low-dimensional Clients

Next we consider the case when X has bounded doubling dimension ddim and Y not. One may wonder if the same approach for the case when Y is low-dimensional applies here. However, due to the asymmetric nature of the problem it is unclear how we could benefit from the dimension reduction in this case. Intuitively, this can be seen by the fact that "moving" a facility/center can be much more costly than moving a client. Indeed, if we move a point in X by a distance z we change the cost of any solution by at most z. Moving a point in Y can change the cost of a solution by as much as $|X| \cdot z$, since we could potentially assign every point in X to the same point in Y. Thus, it is unclear how to apply the previous approach.

Our new decomposition. For the reason above, we extend Talwar's hierarchical decomposition to the case that only a subset of the metric space is doubling. We start from constructing Talwar's decomposition on top of X; denote the resulting decomposition by $\mathcal{H}$. It then remains to decide how to add the points in Y to $\mathcal{H}$. Intuitively, we want that every point $y \in Y$ always lies in the same cluster as $\pi_X(y)$, the nearest neighbor of y in X. More concretely, our plan is to assign to every $y \in Y$ a suitable level $0 \leq h(y) \leq L$, and add y to $\mathcal{H}$ in such a way that

1. $\{y\}$ is a leaf node at level $h(y)$;
2. y and $\pi_X(y)$ are in the same cluster at levels higher than (or equal to) $h(y) + 1$.

We call such cluster $\{y\}$ an *ornament* at level $h(y)$. After adding all $y \in Y$ to $\mathcal{H}$, we obtain a hierarchical decomposition for $X \cup Y$, denoted by $\mathcal{P}$.

It could be tricky to define for every ornament $\{y\}$ the level $h(y)$ to which it should be attached as a leaf. On the one hand, we want $h(y)$ to be sufficiently large, so that y can be covered by some level $h(y)$ portal (i.e., $\mathbf{d}(y, P_C) \le \varepsilon 2^{h(y)}$ for the cluster $C \ni y$). Therefore, we can extend the definition of portal-respecting path to $\mathcal{P}$, and define portal-respecting distance $\mathbf{d}^{\mathcal{P}}_{\mathrm{port}}(x, y)$ on $\mathcal{P}$ the same way as $\mathbf{d}^{\mathcal{H}}_{\mathrm{port}}(x, y)$. On the other hand, we want $h(y)$ to be sufficiently small, so that the error incurred by $\mathbf{d}^{\mathcal{P}}_{\mathrm{port}}(x, y)$, namely $\mathbf{d}^{\mathcal{P}}_{\mathrm{port}}(x, y) - \mathbf{d}(x, y) = O(\varepsilon)2^{h(y)}$, is negligible.

As a first attempt, consider choosing $h(y) = \log(\mathbf{d}(y, X)/\varepsilon)$. Then we have $\mathbf{d}(y, \pi_X(y)) \le \varepsilon 2^{h(y)}$, and we can further argue that y is within distance $\varepsilon 2^{h(y)+1}$ to some level $h(y) + 1$ portal. Therefore, the portal-respecting distance $\mathbf{d}^{\mathcal{P}}_{\mathrm{port}}(x, y)$ can be defined the same way as $\mathbf{d}^{\mathcal{H}}_{\mathrm{port}}(x, y)$. However, this choice of $h(y)$ becomes problematic in terms of the error $\mathbf{d}^{\mathcal{P}}_{\mathrm{port}}(x, y) - \mathbf{d}(x, y)$. Consider an arbitrary point $x \in X$, and assume $h(y)$ is the highest level where $\{x, y\}$ is cut. Then the error is $\mathbf{d}^{\mathcal{P}}_{\mathrm{port}}(x, y) - \mathbf{d}(x, y) = O(\varepsilon)2^{h(y)} = O(\mathbf{d}(y, X))$, which is too large to afford.

To resolve the issue, we choose $h(y)$ as a slightly smaller value $\log(\mathbf{d}(y, X)/\sqrt{\varepsilon})$. For this choice, the error becomes $\mathbf{d}^{\mathcal{P}}_{\mathrm{port}}(x, y) - \mathbf{d}(x, y) = O(\varepsilon)2^{h(y)} = O(\sqrt{\varepsilon})\,\mathbf{d}(y, X)$, which is at most $O(\sqrt{\varepsilon})\,\mathbf{d}(x, y)$ and thus can be charged to $\mathbf{d}(x, y)$. The tradeoff is that y is no longer $\varepsilon 2^{h(y)+1}$-covered by the level $h(y) + 1$ portal set. Instead, the covering radius becomes $\sqrt{\varepsilon}2^{h(y)+1}$. Nonetheless, we can still define $\mathbf{d}^{\mathcal{P}}_{\mathrm{port}}(x, y)$ similarly under this weaker covering property, and it does not change the $O(\sqrt{\varepsilon})\,\mathbf{d}(x, y)$ error bound above when the cutting level is exactly $h(y)$. We obtain a weaker error bound of $\mathbf{d}^{\mathcal{P}}_{\mathrm{port}}(x, y) - \mathbf{d}(x, y) = O(\sqrt{\varepsilon})2^{\ell}$ only if the cutting level ℓ is strictly greater than $h(y)$. In this case, the question reduces to finding the highest cutting level of $\{x, \pi_X(y)\}$, which can be answered fairly well using techniques in [16] and the previous section, since both points are in X.

Dynamic program. Our new hierarchical decomposition $\mathcal{P}$ and the portal-respecting distance $\mathbf{d}^{\mathcal{P}}_{\mathrm{port}}(x, y)$ can then be combined with the dynamic program of Cohen-Addad et al. [16] to get our result in the case when the clients are low-dimensional. This seems difficult at first, because each cluster on $\mathcal{P}$ now has an unbounded number of child clusters, mainly due to the newly added ornaments. A naive enumeration of the configurations of these ornament children would blow up the time complexity of the DP.

Perhaps surprisingly, we show that we can avoid doing the enumeration for ornaments, and it suffices to only enumerate the configurations of non-ornament children, the number of which is bounded by $2^{O(\mathrm{ddim})}$. Our key observation is that ornaments must be candidate facilities, and thus only the opening cost needs to be computed. Therefore, we can obtain from portals of non-ornament children the information which ornaments are potentially required to be opened as a facility. Once we have this information, we simply select the smallest set of ornaments that serve all the unserved portals. This can be done via solving a set cover problem with a bounded universe.

4.3 Applications to the (k, ℓ)-Median Problem under Discrete Fréchet Distance

We next discuss how to apply our results to the (k, ℓ)-median problem under discrete Fréchet distance. It is a folklore result that the doubling dimension of the metric space $(\mathbb{X}^d_z, \mathbf{d}_{dF})$, i.e. the space of polygonal curves of z points in d-dimensional space equipped with the discrete

Fréchet distance[1], is $\Theta(dz)$. This implies that for the (k,ℓ)-median problem the space of center candidates has bounded doubling dimension (for constant ℓ and d). However, it also has an infinite number of points. Thus, we have to compute a discrete subset of that space that contains a $(1+\varepsilon)$-approximation and is not too large. In order to do so we make use of a result by Filtser et al. [24] from the context of nearest neighbor search. Applying Theorem 3, we obtain the following result.

▶ **Theorem 5.** *There is a randomized algorithm that, given as input $\varepsilon \in (0,\frac{1}{2})$, $n,z,k,\ell \in \mathbb{N}$, $P \subset \mathbb{X}_z^d$ with $|P| = n$ computes a $(1+\varepsilon)$-approximate solution to the (k,ℓ)-median problem in time $2^{2^t} \cdot \tilde{O}(d\ell nz)$ with constant success probability, where $t \in O(d\ell \log(d\ell/\varepsilon))$.*

Complexity reduction. For the case of the ambient space being one-dimensional we develop a new complexity reduction for the discrete Fréchet distance that can be summarized as follows. This provides an alternative approach that allows to apply the work of Cohen-Addad et al. [16] in a more direct way for this special case.

We believe that this complexity reduction is of independent interest. For example, it allows us to get an improved coreset construction for clustering under the discrete Fréchet distance.

▶ **Theorem 7** (Complexity reduction). *Let $\varepsilon \in (0,1)$ and $\ell \in \mathbb{N}$ be constants. There is an algorithm that, given an input time series $x \in \mathbb{R}^z$, computes in time $O(z\ell\log^2 z) + O(\ell/\varepsilon)^{2z'}$ with $z' \in O(2^{O(\ell/\varepsilon)^{(2\ell+2)}})$ a time series $x' \in \mathbb{R}^{z'}$ s.t. for all $y \in \mathbb{R}^\ell$,*

$$(1-\varepsilon)\,\mathbf{d}_{dF}(x,y) \le \mathbf{d}_{dF}(x',y) \le (1+\varepsilon)\,\mathbf{d}_{dF}(x,y).$$

Our idea for the complexity reduction can be described as follows. As a first (simple) step, we show that one can reduce the number of distinct values appearing in a time series of complexity z to $O(\ell/\varepsilon)$ while maintaining the distance to any time series of complexity ℓ up to a factor of $(1+\varepsilon)$.

Then we observe that the discrete Fréchet distance between two time series x and y, each of fixed complexity, can be written as a minimum over all traversals. Our goal is to describe the function minimizing over all traversals with a function minimizing over a much smaller set. During each traversal, every value y_i of y is matched to a subsequence of x. In order to determine the Fréchet distance, it suffices to know the minimum and maximum value of x matched to y_i. As it turns out, each traversal can equivalently (but not uniquely!) be described by remembering a sequence of constraints that consist of the minimum and maximum value matched to each y_i. Furthermore, the function minimizing over the set of all possible traversals can likewise be described by minimizing over all possible ordered constraint sets. Since we have reduced the number of different values of the time series to $O(\ell/\varepsilon)$, the number of different constraint sets is small. The set of constraints will be called an ℓ-profile. It is important to note that the set of all ℓ-profiles of a time series x with values from a fixed set X completely determines the Fréchet distance to any time series of complexity ℓ. Since there are only a constant number of different sets of ℓ-profiles (where the constant depends on ℓ and ε) for any time series x with $O(\ell/\varepsilon)$ distinct values, we can replace x by the shortest time series x' over the same set of values that has the same set of ℓ-profiles. The length of the shortest such time series is a constant that depends on the set

[1] Technically, we consider equivalence classes of curves of pairwise discrete Fréchet distance 0 to obtain a proper metric space.

of profiles and the number of distinct values of the time series. Since the number of profiles is also a constant depending on ε and ℓ, the maximum length of these shortest time series is constant as well. We can compute such a time series using a dynamic programming approach. The resulting time series have length $O(2^{O(\ell/\varepsilon)^{(2\ell+2)}})$.

5 Further Related Work

The k-median problem in metric spaces is known to be hard to approximate with a factor better than $1+2/e$ unless set cover can be approximated within a factor $c \ln n$ for $c < 1$ [32]. A number of different constant factor polynomial time approximation algorithms are known [12, 33, 4, 27, 18, 38, 13, 42], and the currently best approximation ratio is $(2+\varepsilon)$ [17]. In the Euclidean plane, the problem is NP-hard [37]. The first polynomial time approximation scheme for k-median in the Euclidean plane has been developed by Arora et al. [3] and later improved to near-linear time in $\mathbb{R}^d$ when d is constant [34]. This result has been generalized to metric spaces of bounded doubling dimension [42] and later to a near-linear approximation scheme [16].

The metric (uncapacitated) facility location problem can be approximated within a constant factor [40, 11, 33, 38, 42, 36, 9], and the currently best polynomial time approximation algorithm achieves an approximation guarantee of 1.488 [35]. At the same time, there is a conditional lower bound of 1.463 on the best possible approximation [26, 32]. In the constant-dimensional Euclidean setting, there are similar results as for the k-median problem [34, 41, 16].

Driemel et al. [21] defined the (k, ℓ)-clustering problem for time series as follows: Given a set P of n time series of complexity m and parameters $k, \ell \in \mathbb{N}$ find k center time series of complexity ℓ, such that (a) the maximum distance of an element in P to its closest center time series or (b) the sum of these distances is minimized. Variant (a) is referred to as (k, ℓ)-center and (b) as (k, ℓ)-median. Under the continuous Fréchet distance, they developed near-linear time $(1+\varepsilon)$-approximation algorithms for both clustering variants, assuming ε, k, and ℓ are constants. They complement these algorithmic results with hardness results, showing that both (k, ℓ)-median and (k, ℓ)-center are NP-hard under continuous Fréchet distance. Approximating (k, ℓ)-median for polygonal curves in arbitrary dimensions was recently studied in [7]. Cheng and Huang give the first $(1+\varepsilon)$-approximation algorithm for (k, ℓ)-median under continuous Fréchet distance in $d > 1$ [14]. Both clustering problems are also NP-hard under the discrete Fréchet distance and even for the case $k = 1$ [5] [6]. Buchin et al. developed the first $(1+\varepsilon)$-approximation algorithm for (k, ℓ)-median under discrete Fréchet distance, which runs in $\tilde{O}((1/\varepsilon k\ell)^{k\ell} \cdot k\ell n m^{(k\ell+1)})$ time [6]. Nath and Taylor [39] improved this to $\tilde{O}(nm2^{O(k/\varepsilon \log(k/\varepsilon))} \cdot (\ell/\varepsilon^2)^{O(kl)})$. Buchin and Rohde [8] designed the first coreset construction for (k, ℓ)-median under both variants of the Fréchet distance, where the size of the coreset has logarithmic dependence on the number of input curves. Recently, Cohen-Addad et al. introduced a coreset construction for (k, ℓ)-median under discrete Fréchet distance that has size independent of the number of input curves [15].

Related to our dimension reduction are some data structures for approximate nearest neighbor search under discrete Fréchet distance [22, 23, 24]. In the asymmetric setting where the query time series has complexity ℓ, the data structures cited above replace each input time series by a set of lower dimensional time series. This is fundamentally different from our dimension reduction, which replaces each time series with exactly one lower dimensional time series.

References

1 Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 271–286. ACM, 2006. doi:10.1145/1132516.1132557.

2 Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998. doi:10.1145/290179.290180.

3 Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. Approximation Schemes for Euclidean k-Medians and Related Problems. In *30th Annual ACM Symposium on the Theory of Computing*, pages 106–113, 1998. doi:10.1145/276698.276718.

4 Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local Search Heuristics for k-Median and Facility Location Problems. In *SIAM Journal on Computing*, volume 33, pages 544–562, 2004. doi:10.1137/S0097539702416402.

5 Kevin Buchin, Anne Driemel, Joachim Gudmundsson, Michael Horton, Irina Kostitsyna, Maarten Löffler, and Martijn Struijs. Approximating (k, ℓ)-center clustering for curves. In *30th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2922–2938, 2019. doi:10.1137/1.9781611975482.181.

6 Kevin Buchin, Anne Driemel, and Martijn Struijs. On the Hardness of Computing an Average Curve. In *17th Scandinavian Symposium and Workshops on Algorithm Theory*, volume 162, pages 19:1–19:19, 2020. doi:10.4230/LIPIcs.SWAT.2020.19.

7 Maike Buchin, Anne Driemel, and Dennis Rohde. Approximating (k,ℓ)-Median Clustering for Polygonal Curves. In *ACM Transactions on Algorithms*, volume 19, pages 4:1–4:32, 2023. doi:10.1145/3559764.

8 Maike Buchin and Dennis Rohde. Coresets for (k,ℓ)-Median Clustering Under the Fréchet Distance. In *Algorithms and Discrete Applied Mathematics*, pages 167–180, 2022. doi:10.1007/978-3-030-95018-7_14.

9 Jaroslaw Byrka and Karen Aardal. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. *SIAM Journal on Computing*, 39(6):2212–2231, 2010. doi:10.1137/070708901.

10 T.-H. Hubert Chan, Shuguang Hu, and Shaofeng H.-C. Jiang. A PTAS for the steiner forest problem in doubling metrics. *SIAM J. Comput.*, 47(4):1705–1734, 2018. doi:10.1137/16M1107206.

11 Moses Charikar and Sudipto Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*, pages 378–388. IEEE, 1999. doi:10.1109/SFFCS.1999.814609.

12 Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A Constant-Factor Approximation Algorithm for the k-Median Problem. In *Journal of Computer and System Sciences*, volume 65, pages 129–149, 2002. doi:10.1006/JCSS.2002.1882.

13 Moses Charikar and Shi Li. A Dependent LP-Rounding Approach for the k-Median Problem. In *Automata, Languages, and Programming - 39th International Colloquium*, pages 194–205, 2012. doi:10.1007/978-3-642-31594-7_17.

14 Siu-Wing Cheng and Haoqiang Huang. Curve Simplification and Clustering under Fréchet Distance. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1414–1432, 2023. doi:10.1137/1.9781611977554.CH51.

15 Vincent Cohen-Addad, Andrew Draganov, Matteo Russo, David Saulpic, and Chris Schwiegelshohn. A tight vc-dimension analysis of clustering coresets with applications. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 4783–4808. SIAM, 2025. doi:10.1137/1.9781611978322.162.

16 Vincent Cohen-Addad, Andreas Emil Feldmann, and David Saulpic. Near-linear time approximation schemes for clustering in doubling metrics. *J. ACM*, 68(6):44:1–44:34, 2021. doi:10.1145/3477541.

17 Vincent Cohen-Addad, Fabrizio Grandoni, Euiwoong Lee, Chris Schwiegelshohn, and Ola Svensson. A (2+ε)-approximation algorithm for metric k-median. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 615–624. ACM, 2025. doi:10.1145/3717823.3718299.

18 Vincent Cohen-Addad, Anupam Gupta, Lunjia Hu, Hoon Oh, and David Saulpic. An Improved Local Search Algorithm for k-Median. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1556–1612, 2022. doi:10.1137/1.9781611977073.65.

19 Anne Driemel, Jan Höckendorff, Ioannis Psarros, Christian Sohler, and Di Yue. Near linear time approximation schemes for clustering of partially doubling metrics. *CoRR*, 2026. doi:10.48550/arXiv.2603.24336.

20 Anne Driemel, Jan Höckendorff, Ioannis Psarros, and Christian Sohler. A near-linear time approximation scheme for (k, ℓ)-median clustering under discrete fréchet distance, 2025. doi:10.48550/arXiv.2508.07008.

21 Anne Driemel, Amer Krivosija, and Christian Sohler. Clustering time series under the Fréchet distance. In *27th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 766–785, 2016. doi:10.1137/1.9781611974331.ch55.

22 Anne Driemel, Ioannis Psarros, and Melanie Schmidt. Sublinear data structures for short Fréchet queries. In *Computing Research Repository*, 2019. arXiv:1907.04420.

23 Arnold Filtser and Omrit Filtser. Static and Streaming Data Structures for Fréchet Distance Queries. In *ACM Transactions on Algorithms*, volume 19, pages 39:1–39:36, 2023. doi:10.1145/3610227.

24 Arnold Filtser, Omrit Filtser, and Matthew J. Katz. Approximate Nearest Neighbor for Curves: Simple, Efficient, and Deterministic. In *Algorithmica*, volume 85, pages 1490–1519, 2023. doi:10.1007/S00453-022-01080-1.

25 Lee-Ad Gottlieb and Robert Krauthgamer. Proximity algorithms for nearly doubling spaces. *SIAM J. Discret. Math.*, 27(4):1759–1769, 2013. doi:10.1137/120874242.

26 Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of algorithms*, 31(1):228–248, 1999. doi:10.1006/JAGM.1998.0993.

27 Anupam Gupta and Kanat Tangwongsan. Simpler Analyses of Local Search Algorithms for Facility Location. In *Computing Research Repository*, 2008. arXiv:0809.2554.

28 Sariel Har-Peled and Nirman Kumar. Approximate nearest neighbor search for low-dimensional queries. *SIAM J. Comput.*, 42(1):138–159, 2013. doi:10.1137/110852711.

29 Sariel Har-Peled and Manor Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006. doi:10.1137/S0097539704446281.

30 Lingxiao Huang, Shaofeng H.-C. Jiang, Robert Krauthgamer, and Di Yue. Near-optimal dimension reduction for facility location. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 665–676. ACM, 2025. doi:10.1145/3717823.3718214.

31 Piotr Indyk and Assaf Naor. Nearest-neighbor-preserving embeddings. *ACM Trans. Algorithms*, 3(3):31–es, August 2007. doi:10.1145/1273340.1273347.

32 Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 731–740, New York, NY, USA, 2002. Association for Computing Machinery. doi:10.1145/509907.510012.

33 Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-Median problems using the primal-dual schema and Lagrangian relaxation. In *Journal of the ACM*, volume 48, pages 274–296, 2001. doi:10.1145/375827.375845.

34 Stavros G. Kolliopoulos and Satish Rao. A Nearly Linear-Time Approximation Scheme for the Euclidean k-Median Problem. In *SIAM Journal on Computing*, volume 37, pages 757–782, 2007. doi:10.1137/S0097539702404055.

35 Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. In Luca Aceto, Monika Henzinger, and Jirí Sgall, editors, *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part II*, volume 6756 of *Lecture Notes in Computer Science*, pages 77–88. Springer, 2011. doi:10.1007/978-3-642-22012-8_5.

36 Mohammad Mahdian, Yinyu Ye, and Jiawei Zhang. Approximation algorithms for metric facility location problems. *SIAM Journal on Computing*, 36(2):411–432, 2006. doi:10.1137/S0097539703435716.

37 Nimrod Megiddo and Kenneth J. Supowit. On the Complexity of Some Common Geometric Location Problems. In *SIAM Journal on Computing*, volume 13, pages 182–196, 1984. doi:10.1137/0213014.

38 Ramgopal R. Mettu and C. Greg Plaxton. The Online Median Problem. In *SIAM Journal on Computing*, volume 32, pages 816–832, 2003. doi:10.1137/S0097539701383443.

39 Abhinandan Nath and Erin Taylor. k-Median clustering under discrete Fréchet and Hausdorff distances. In *Journal of Computational Geometry*, volume 12, pages 156–182, 2021. doi:10.20382/JOCG.V12I2A8.

40 David B Shmoys, Éva Tardos, and Karen Aardal. Approximation algorithms for facility location problems. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 265–274, 1997.

41 Kunal Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 281–290. ACM, 2004. doi:10.1145/1007352.1007399.

42 Mikkel Thorup. Quick k-Median, k-Center, and Facility Location for Sparse Graphs. In *SIAM Journal on Computing*, volume 34, pages 405–432, 2004. doi:10.1137/S0097539701388884.