Local Search Yields a PTAS for k-Means in Doubling Metrics *

Zachary Friggstad[†] Mohsen Rezapour Mohammad R. Salavatipour[‡]

Department of Computing Science
University of Alberta

Abstract

The most well known and ubiquitous clustering problem encountered in nearly every branch of science is undoubtedly k-MEANS: given a set of data points and a parameter k, select k centres and partition the data points into k clusters around these centres so that the sum of squares of distances of the points to their cluster centre is minimized. Typically these data points lie in Euclidean space $\mathbb{R}^d$ for some $d \geq 2$.

k-MEANS and the first algorithms for it were introduced in the 1950's. Over the last six decades, hundreds of papers have studied this problem and different algorithms have been proposed for it. The most commonly used algorithm in practice is known as Lloyd-Forgy, which is also referred to as "the" k-MEANS algorithm, and various extensions of it often work very well in practice. However, they may produce solutions whose cost is arbitrarily large compared to the optimum solution. Kanungo et al. [2004] analyzed a very simple local search heuristic to get a polynomial-time algorithm with approximation ratio $9 + \epsilon$ for any fixed $\epsilon > 0$ for k-MEANS in Euclidean space.

Finding an algorithm with a better worst-case approximation guarantee has remained one of the biggest open questions in this area, in particular whether one can get a true PTAS for fixed dimension Euclidean space. We settle this problem by showing that a simple local search algorithm provides a PTAS for k-MEANS for $\mathbb{R}^d$ for any fixed d.

More precisely, for any error parameter $\epsilon > 0$, the local search algorithm that considers swaps of up to $\rho = d^{O(d)} \cdot \epsilon^{-O(d/\epsilon)}$ centres at a time will produce a solution using *exactly* k centres whose cost is at most a $(1 + \epsilon)$-factor greater than the optimum solution. Our analysis extends very easily to the more general settings where we want to minimize the sum of q'th powers of the distances between data points and their cluster centres (instead of sum of squares of distances as in k-MEANS) for any fixed $q \geq 1$ and where the metric may not be Euclidean but still has fixed doubling dimension.

Finally, our techniques also extend to other classic clustering problems. We provide the first demonstration that local search yields a PTAS for UNCAPACITATED FACILITY LOCATION and the generalization of k-MEDIAN to the setting with non-uniform opening costs in doubling metrics.

1 Introduction

With advances in obtaining and storing data, one of the emerging challenges of our age is data analysis. It is hard to find a scientific research project which does not involve some form of methodology

*A preliminary version of this paper appeared in Proceedings of 57th FOCS 2016 [27]

†This research was undertaken, in part, thanks to funding from the Canada Research Chairs program and an NSERC Discovery Grant.

‡Supported by NSERC.

to process, understand, and summarize data. A large portion of data analysis is concerned with predicting patterns in data after being trained with some training data set (machine learning). Two problems often encounterd in data analysis are classification and clustering. Classification (which is an instance of supervised learning) is the task of predicting the label of a new data point after being trained with a set of labeled data points (called a training set). Basically, after given a training set of correctly labeled data points the program has to identify the label of a given new (unlabeled) data point. Clustering (which is an instance of unsupervised learning) is the task of grouping a given set of objects or data points into clusters/groups such that the data points that are more similar fall into the same cluster while data points (objects) that do not seem similar are in different clusters. Some of the main purposes of clustering are to understand the underlying structure and relation between objects and find a compact representation of data points.

Clustering and different methods to achieve it have been studied since the 1950's in different branches of science: biology, statistics, medical sciences, computer science, social sciences, engineering, physics, and more. Depending on the notion of what defines a cluster, different models of clustering have been proposed and studied by researchers. Perhaps the most widely used clustering model is the k-MEANS clustering: Given a set $\mathcal{X}$ of n *data points* in d-dimensional Euclidean space $\mathbb{R}^d$, and an integer k, find a set of k points $c_1, \ldots, c_k \in \mathbb{R}^d$ to act as as *centres* that minimize the sum of squared distances of each data point to its nearest centre. In other words, we would like to partition $\mathcal{X}$ into k cluster sets, $\{C_1, \ldots, C_k\}$ and find a centre c_i for each C_i to minimize

$$\sum_{i=1}^{k} \sum_{x \in C_i} ||x - c_i||_2^2.$$

Here, $||x - c_i||_2$ is the standard Euclidean distance in $\mathbb{R}^d$ between points x and c_i.

This value is called the cost of the clustering. Typically, the centres c_i are selected to be the centroid (mean) of the cluster C_i. In other situations the centres must be from the data points themselves (i.e. $c_i \in C_i$) or from a given set $\mathcal{C}$. This latter version is referred to as discrete k-MEANS clustering. Although in most application of k-MEANS the data points are in some Euclidean space, the discrete variant can be defined in general metrics. The k-MEANS clustering problem is known to be an NP-hard problem even for $k = 2$ or when $d = 2$ [1, 45, 23, 50].

Clustering, in particular the k-MEANS clustering problem as the most popular model for it, has found numerous applications in very different areas. The following is a (short) list of applications of clustering that have been addressed by Jain [35]: image segmentation, information access, grouping customers into different types for efficient marketing, grouping delivery services for workforce management and planning, and grouping genome data in biology. For instance, clustering is used to identify groups of genes with related expression patterns in a key step of the analysis of gene functions and cellular processes. It is also extensively used to group patients based on their genetic, pathological, and cellular features which is proven useful in analyzing human genetics diseases (e.g., see [22, 33]).

The most widely used algorithm for k-MEANS (which is also sometimes referred to as "the" k-means algorithm) is a simple heuristic introduced by Lloyd in 1957 [44]. This algorithm starts from an initial partition of the points into k clusters and it repeats the following two steps as long as it improves the quality of the clustering: pick the centroids of the clusters as centres, and then re-compute a new clustering by assigning each point to the nearest centre. Although this algorithm works well in practice it is known that the ratio of the cost of the solution computed by this algorithm vs the optimum solution cost (known as the "approximation ratio") can be arbitrarily large (see [36]). Various modifications and extensions of this algorithm have been produced and studied, e.g. ISODATA, FORGY, Fuzzy C-means, k-means++, filtering using kd-trees (see [35]), but none of them are known to have a bounded approximation ratio in the general setting. Arthur and Vassilvitskii [6] show that Lloyd's method with properly chosen initial centres will be an $O(\log k)$-approximation. Ostrovsky et al. [48] show that under some assumptions about the data points the approximation ratio is bounded by a constant. The problem of finding an efficient algorithm for k-MEANS with a proven theoretical bound on the cost of the solution returned is probably one of the most well studied problems in the whole field of clustering with hundreds of research papers devoted to this.

Arthur and Vassilvitskii [5], Dastupta and Gupta [20], and Har-Peled and Sadri [32] study convergence rate of Lloyd's algorithm. In particular [5] show that it can be super-polynomial. More recently, Vattani [51] shows that it can take exponential time even in two dimensions. Arthur et al. [4] proved that Lloyd's algorithm has polynomial-time smoothed complexity. Kumar and Kannan [38], Ostrovsky et al. [48], and Awasthi et al. [9] gave empirical and theoretical evidence for when and why the known heuristics work well in practice. For instance [9] show that when the size of an optimum $(k-1)$-MEANS is sufficiently larger than the cost of k-MEANS then one can get a near optimum solution to k-MEANS using a variant of Lloyd's algorithm.

The k-MEANS problem is known to be NP-hard [1, 23, 45]. In fact, the k-MEANS problem is NP-hard if d is arbitrary even for $k = 2$ [1, 23]. Also, if k is arbitrary the problem is NP-hard even for $d = 2$ [45, 50]. However, the k-MEANS problem can be solved in polynomial time by the algorithm of [34] when both k and d are constant.

A *polynomial-time approximation scheme* (PTAS) is an algorithm that accepts an additional parameter $\epsilon > 0$. It finds solutions whose cost is at most $1+\epsilon$ times the optimum solution cost and runs in polynomial time when ϵ can be regarded as a fixed constant (i.e. $O(n^{f(\epsilon)})$ for some function f). Matoušek [46] gave a PTAS for fixed k, d with running time $O(n(\log n)^k \epsilon^{-2k^2 d})$. Since then several other PTASs have been proposed for variant settings of parameters but all need k to be constant [47, 11, 21, 31, 39, 40, 25, 30]. Bādoiu et al. [11] gave a PTAS for fixed k and any d with time $O(2^{(k/\epsilon)^{O(1)}} \mathrm{poly}(d) n \log^k n)$. De la Vega et al. [21] proposed a $(1+\epsilon)$-approximation with running time $O(2^{(k^3/\epsilon^8)(\ln(k/\epsilon))\ln k} dn \log^k n)$. This was improved to $O(2^{(k/\epsilon)^{O(1)}} dn)$ by Kumar et al. [39, 40]. By building *coresets* of size $O(k\epsilon^{-d} \log n)$, Har-Peled and Mazumdar [31] presented a $(1+\epsilon)$-approximation with time $O(n + k^{k+1}\epsilon^{-(2d+1)k} \log^{k+1} n \log^k \frac{1}{\epsilon})$. This was slightly improved by Har-Peled and Kushal [30]. Feldman et al. [25] developed another $(1+\epsilon)$-approximation with running time $\tilde{O}(nkd + d\cdot\mathrm{poly}(k/\epsilon) + 2^{\tilde{O}(k/\epsilon)})$, where the $\tilde{O}(.)$ notation hides polylog factors. Recently, Bandyapadhyay and Varadarajan [12] presented a pseudo-approximation for k-MEANS in fixed-dimensional Euclidean space: their algorithm finds a solution whose cost is at most $1+\epsilon$ times of the optimum but might use up to $(1+\epsilon)\cdot k$ clusters.

The result of Matoušek [46] also shows that one can select a set $\mathcal{C}$ of "candidate" centers in $\mathbb{R}^d$ from which the k centres should be chosen from with a loss of at most $(1+\epsilon)$ and this set can be computed in time $O(n\epsilon^{-d}\log(1/\epsilon))$. This reduces the k-means problem to the discrete setting where along with $\mathcal{X}$ we have a set $\mathcal{C}$ of candidate centres and we have to select k centres from $\mathcal{C}$. Kanungo et al. [36] proved that a simple local search heuristic yields an algorithm with approximation ratio $9+\epsilon$ for $\mathbb{R}^d$. This remains the best known approximation algorithm with polynomial running time for $\mathbb{R}^d$. They also present an algorithm which is a hybrid of local search and Lloyd algorithm and give empirical evidence that this works much better in practice (better than Lloyd's algorithm) and has proven constant factor ratio. For general metrics, Gupta and Tangwongsan [29] proved that local search is a $(25+\epsilon)$-approximation. It was an open problem for a long time whether k-MEANS is APX-hard or not in Euclidean metrics. This was recently answered positively by Awasthi et al. [10] where they showed that the problem is APX-hard in $\mathbb{R}^d$ but the dimension d used in the proof is $\Omega(\log n)$. Blomer et al. [13] have a nice survey of theoretical analysis of different k-means algorithms.

k-median
Another very well studied problem that is also closely related to k-MEANS is k-MEDIAN. The only difference is that the goal (objective function) in k-MEDIAN is to minimize the sum of distances, instead of sum of square of distances as in k-MEANS, i.e. minimize $\sum_{i=1}^{k} \sum_{x \in C_i} \delta(x, c_i)$ where $\delta(x, c_i)$ is the distance between x and c_i. This problem occurs in operations research settings.

There are constant factor approximation algorithms for k-MEDIAN in general metrics. The simple local search (which swaps in and out a constant number of centres in each iteration) is known to give a $3 + \epsilon$ approximation by Arya et al. [7, 8]. The current best approximation uses different techniques and has an approximation ratio of $2.611 + \epsilon$ [43, 14]. The local search $(9 + \epsilon)$-approximation (for k-MEANS) in [36] can be seen as an extension of the analysis in [7] for k-MEDIAN. One reason that analysis of k-MEANS is more difficult is that the squares of distances do not necessarily satisfy the triangle inequality. For instances of k-MEDIAN on Euclidean metrics, Arora et al. [3], building on the framework of Arora [2], gave the first PTAS. Kolliopoulos and Rao [37] improved the time complexity and presented a PTAS for Euclidean k-MEDIAN with time complexity $O(2^{O((1+\log \frac{1}{\epsilon})/\epsilon)^{d-1}} n \log n \log k)$. Such an approximation is also known as an *efficient PTAS*: the running time of the $(1 + \epsilon)$-approximation is of the form $f(\epsilon) \cdot \text{poly}(n)$ (in fixed-dimension metrics).

Uncapacitated facility location
The UNCAPACITATED FACILITY LOCATION problem is the same as k-MEDIAN except instead of a cardinality constraint bounding the number of open facilities, we are instead given opening costs f_i for each $i \in \mathcal{C}$. The goal is to find a set of centers $\mathcal{S} \subseteq \mathcal{C}$ that minimizes $\sum_{x \in \mathcal{X}} \delta(x, \mathcal{S}) + \sum_{i \in \mathcal{S}} f_i$. Currently the best approximation for UNCAPACITATED FACILITY LOCATION in general metrics is a 1.488-approximation [42]. As with k-MEDIAN, a PTAS is known for UNCAPACITATED FACILITY LOCATION in constant-dimensional Euclidean metrics [3, 37], with the latter giving an efficient PTAS.

1.1 Our result and technique

Although a PTAS for k-MEDIAN in fixed dimension Euclidean space has been known for almost two decades, getting a PTAS for k-MEANS in fixed dimension Euclidean space has remained an open problem. We provide a PTAS for this setting. We focus on the discrete case where we have to select a set of k centres from a given set $\mathcal{C}$. Our main result is to show that a simple local search heuristic that swaps up to $\rho = d^{O(d)} \cdot \epsilon^{-O(d/\epsilon)}$ centres at a time and assigns each point to the nearest centre is a PTAS for k-MEANS in $\mathbb{R}^d$.

A precise description of the algorithm is given in Section 2. At a high level, we start with any set of k centres $\mathcal{S} \subseteq \mathcal{C}$. Then, while there is some other set of k centres $\mathcal{S}' \subseteq \mathcal{C}$ with $|\mathcal{S} - \mathcal{S}'| \leq \rho$ for some constant ρ such that $\mathcal{S}'$ is a cheaper solution than $\mathcal{S}$, we set $\mathcal{S} \leftarrow \mathcal{S}'$. Repeat until $\mathcal{S}$ cannot be improved any further. Each iteration takes $|\mathcal{C}|^{O(\rho)}$ time, which is polynomial when ρ is a constant. Such a solution is called a *local optimum* solution with respect to the ρ-swap heuristic. We still have to ensure that the algorithm only iterates a polynomial number of times; a standard modification discussed in Section 2 ensures this.

Recall that the *doubling dimension* of a metric space is the smallest τ such that any ball of radius $2r$ around a point can be covered by at most 2^τ balls of radius r. If the doubling dimension can be regarded as a constant then we call the metric a *doubling metric* (as in [49]). The Euclidean metric over $\mathbb{R}^d$ has doubling dimension $O(d)$. Our analysis implies a PTAS for more general settings where the data points are in a metric space with constant doubling dimension (described below) and when the objective function is to minimize the sum of q'th power of the distances for some fixed $q \geq 1$.

Let $\rho(\epsilon, d) := d^{O(d)} \cdot \epsilon^{O(-d/\epsilon)}$. We will articulate the absolute constants suppressed by the $O(\cdot)$ notation later on in our analysis.

Theorem 1 *The local search algorithm that swaps up to $\rho(\epsilon, d)$ centres at a time is a $(1+\epsilon)$-approximation for k-MEANS in metrics with doubling dimension d.*

Now consider the generalization where the objective function measures the sum of q'th power of the distances between points and their assigned cluster centre, we call this ℓ_q^q-NORM k-CLUSTERING. Here, we are given the points $\mathcal{X}$ in a metric space $\delta(\cdot, \cdot)$ along with a set $\mathcal{C}$ of potential centres. We are to select k centres from $\mathcal{C}$ and partition the points into k cluster sets $C_1, \ldots, C_k$ with each C_i having a centre c_i so as to minimize

$$\sum_{i=1}^{k} \sum_{x \in C_i} \delta(x, c_i)^q.$$

Note that the case of $q = 2$ is the k-MEANS problem and $q = 1$ is k-MEDIAN. We note that our analysis extends to provide a PTAS for this setting when q is fixed. That is, Theorem 1 holds for ℓ_q^q-NORM k-CLUSTERING for constants $q \geq 1$, except that we require that the local search procedure be the ρ'-swap heuristic where $\rho' = d^{O(d)} \cdot (2^q/\epsilon)^{O(2^q \cdot d/\epsilon)}$.

Note that even for the case of k-MEDIAN, this is the first PTAS for metrics with constant doubling dimension. Also, while a PTAS was known for k-MEDIAN for constant-dimensional Euclidean metrics, determining if local search provided such a PTAS was an open problem. For example, [18] shows that local search can be used to get a $1+\epsilon$ approximation for k-MEDIAN that uses up to $(1+\epsilon) \cdot k$ centres.

For the ease of exposition, we initially present the proof restricted to k-MEANS in $\mathbb{R}^d$. We explain in Section 5.1 how to extend the analysis to doubling metrics and describe in Section 5.2 how this can be easily extended to prove Theorem 1 when the objective is ℓ_q^q-NORM k-CLUSTERING.

As mentioned earlier, Awasthi et al. [10] proved that k-MEANS is APX-hard for $d = \Omega(\log n)$ and they left the approximability of k-MEANS for lower dimensions as an open problem. A consequence of our algorithm is that one can get a $(1+\epsilon)$-approximation for k-MEANS that runs in sub-exponential time for values of d up to $O(\log n / \log \log n)$. More specifically, for any given $0 < \epsilon < 1$ and $d = \sigma \log n / \log \log n$ for sufficiently small absolute constant σ we get a $(1+\epsilon)$-approximation for k-MEANS that runs in time $O(2^{n^\kappa})$, for some constant $\kappa = \kappa(\sigma) < 1$; for $d = O(\log \log n / \log \log \log n)$ we get a *quasi-polytime approximation scheme* (QPTAS). Therefore, our result in a sense shows that the requirement of [10] of $d = \Omega(\log n)$ to prove APX-hardness of k-MEANS is almost tight unless NP $\subseteq DTIME(2^{n^\sigma})$.

The notion of coresets and using them for finding faster algorithms for k-means has been studied extensively (e.g. [31, 30, 15, 25, 26] and references there). A coreset is a small subset of data points (possibly with weights associated to them) such that running the clustering algorithm on them (instead of the whole data set) generates a clustering of the whole data set with approximately good cost. In order to do this, one can go to the discrete case. To use coresets, we need to be able to solve (discrete) k-means in the more general setting where each centre $i \in \mathcal{C}$ has an associated weight $w(i)$, and the cost of assigning a point j to i (if i is selected to be a centre) is $w(i) \cdot \delta(i, j)^2$. Our local search algorithm works for this weighted setting as well. We will show how to use these ideas to improve the running time of the local search algorithm.

Finally, we observe that our techniques easily extend to show a natural local-search heuristic for UNCAPACITATED FACILITY LOCATION is a PTAS. In particular, for the same constant ρ as in Theorem 1 (in fact, it can be slightly smaller) we consider the natural local search algorithm that returns a solution $\mathcal{S} \subseteq \mathcal{C}$ such that $\mathrm{cost}(\mathcal{S}) \leq \mathrm{cost}(\mathcal{S}')$ for all $\mathcal{S}' \subseteq \mathcal{C}$ with both $|\mathcal{S} - \mathcal{S}'| \leq \rho$ and $|\mathcal{S}' - \mathcal{S}| \leq \rho$ (i.e. add and/or drop up to ρ centres in $\mathcal{C}$).

Theorem 2 *The local search algorithm that adds and/or drops up to $\rho(\epsilon, d)$ centres at a time is a $(1+\epsilon)$-approximation for* UNCAPACITATED FACILITY LOCATION *in metrics with doubling dimension d.*

This seems to be the first explicit record of a PTAS for UNCAPACITATED FACILITY LOCATION in doubling metrics, but one was known in constant-dimensional Euclidean metrics [3, 37]. Prior to this work, local search was only known to provide a PTAS for UNCAPACITATED FACILITY LOCATION in constant-dimensional Euclidean metrics if all opening costs are the same [19]. The basic idea why this works is our analysis in Theorem 1 considers test swaps that, overall, swap in each local optimum centre exactly once and swap out each global optimum centre exactly once (i.e. the swaps come from a full partitioning of the local and global optimum). The partitioning we use to obtain these swaps only uses the assumption that precisely k centres are open in a feasible solution at one point, and this step can be safely ignored in the case of UNCAPACITATED FACILITY LOCATION.

Lastly, we consider the common generalization of k-MEDIAN and UNCAPACITATED FACILITY LOCATION where all centres have costs and at most k centres may be chosen (sometimes called the GENERALIZED k-MEDIAN or k-UNCAPACITATED FACILITY LOCATION problem). We consider the local search operation that tries to add and/or drop up to ρ centres at a time as long as the candidate solution being tested includes at most k centres. We get a PTAS in this case as well. To the best of our knowledge, the previous best approximation was a 5-approximation in general metrics, also obtained by local search [29].

Theorem 3 *The local search algorithm that adds and/or drops up to $\rho(\epsilon, d)$ centres at a time (provided the resulting solution has at most k facilities) is a a $(1+\epsilon)$-approximation for* GENERALIZED k-MEDIAN *in metrics with doubling dimension d.*

The results of Theorems 2 and 3 extend to the setting where we are considering the ℓ_q^q-norm for distances instead of the ℓ_1-norm.

Note: Shortly after we announced our result [28], Cohen-Addad, Klein, and Mathieu [17, 16] announced similar results for k-MEANS on Euclidean and minor-free metrics using the local search method. Specifically, they prove that the same local search algorithm yields a PTAS for k-MEANS on Euclidean and minor-free metrics. These results are obtained independently.

1.2 Proof Outline

The general framework for analysis of local search algorithms for k-MEDIAN and k-MEANS in [8, 36, 29] is as follows. Let $\mathcal{S}$ and $\mathcal{O}$ be a local optimum and a global optimum solution, respectively. They carefully identify a set Q of potential swaps between local and global optimum. In each such swap, the cost of assigning a data point x to the nearest centre after a swap is bounded with respect to the local and global cost assignment. In other words, if $\mathcal{B}(\mathcal{S})$ is the set of solutions obtained by performing swaps from Q, the main task is to show that $\sum_{\mathcal{S}' \in \mathcal{B}(\mathcal{S})}(\mathrm{cost}(\mathcal{S}') - \mathrm{cost}(\mathcal{S})) \leq \alpha \cdot \mathrm{cost}(\mathcal{O}) - \mathrm{cost}(\mathcal{S})$ for some constant α. Given that $0 \leq \mathrm{cost}(\mathcal{S}') - \mathrm{cost}(\mathcal{S})$ for all $\mathcal{S}' \in \mathcal{B}(\mathcal{S})$ (because $\mathcal{S}$ is a local optimum), $\mathrm{cost}(\mathcal{S}) \leq \alpha \cdot \mathrm{cost}(\mathcal{O})$.

Our analysis has the same structure but has many more ingredients and several intermediate steps to get us what we want. Note that the following only describes steps used in the analysis of the local search algorithm; we do not perform any of the steps described below in the algorithm itself.

Let us define $\mathcal{S}$ and $\mathcal{O}$ as before. First, we do a filtering over $\mathcal{S}$ and $\mathcal{O}$ to obtain subsets $\overline{\mathcal{S}} \subseteq \mathcal{S}$ and $\overline{\mathcal{O}} \subseteq \mathcal{O}$ such that every centre in $\mathcal{S} - \overline{\mathcal{S}}$ (in $\mathcal{O} - \overline{\mathcal{O}}$) is "close" to a centre in $\overline{\mathcal{S}}$ (in $\overline{\mathcal{O}}$) while these filtered centres are far apart. We define a "net" around each centre $i \in \overline{\mathcal{S}}$ which captures a collection of other filtered centres in $\overline{\mathcal{O}}$ that are relatively close to i. The idea of the net is that if we choose to close i (in a test swap) then the data points that were to be assigned to i will be assigned to a nearby centre in the net of i. Since the metric is a constant-dimensional Euclidean metric (or, more generally, a doubling metric), we can choose these nets to have constant size.

For each j assigned to i in $\mathcal{S}$, if the centre i^* that j is assigned to in the optimum solution lies somewhat close to i then we can reassign j to a facility in the net around i that is close to i^*. In this case, the reassignment cost for j will be close to $c_j^* - c_j$. Otherwise, if i^* lies far from i then we can reassign j to a facility near i in the net around i and the reassignment cost will only be $O(\epsilon) \cdot (c_j^* + c_j)$ and we will generate the $c_j^* - c_j$ term for the local search analysis when i^* is opened in another different swap.

Of course, there are some complications in that we need to do something else with j if the net around i is not open. This will happen infrequently, but we need a somewhat reasonable bound when it does happen. Also, for reasons that will become apparent in the analysis we do something different in the case that i^* is somewhat close to i but i^* is much closer to a different facility in $\mathcal{S}$ than it is to i.

One main part of our proof is to show that there exists a suitable randomized partitioning of $\mathcal{S} \cup \mathcal{O}$ such that each part has small size (which will be a function of only ϵ and d and, ultimately, determines the size of the swaps in our local search procedure), and for any pair $(i, i^*) \in \overline{\mathcal{S}} \times \overline{\mathcal{O}}$ where i^* lies in the net of i we have $\Pr[i, i^* \text{ lie in the same part}] \geq 1 - \epsilon$. This randomized partitioning is the only part of the proof that we rely on properties of doubling metrics (or $\mathbb{R}^d$). For those small portion of facilities that their net is "cut" by our partitioning, we show that when we close those centres in our test swaps then the reassignment cost of a point j that was assigned to them is only $O(1)$ times more than the sum of their assignment costs in $\mathcal{O}$ and $\mathcal{S}$. Given that this only happens with small probability (due to our random partition scheme), this term is negligible in the final analysis. So we get an overall bound of $1 + O(\epsilon)$ on the ratio of cost of $\mathcal{S}$ over $\mathcal{O}$.

Outline of the paper: We start with some basic definitions and notation in Section 2. In Sections 3 and 4 we show that the local search algorithm with an appropriate number of swaps provides a PTAS for k-MEANS in $\mathbb{R}^d$. In Section 5 we show how this can be extended to prove Theorem 1 and to the setting where we measure the ℓ_q^q-norm of the solution for any constant $q \geq 1$. Finally, before concluding, in Section 6 we show that our analysis easily extends to prove Theorems 2 and 3.

2 Notation and Preliminaries

Recall that in the k-MEANS problem we are given a set $\mathcal{X}$ of n points in $\mathbb{R}^d$ and an integer $k \geq 1$; we have to find k centres $c_1, \ldots, c_k \in \mathbb{R}^d$ so as to minimize the sum of squares of distances of each point to the nearest centre. As mentioned earlier, by using the result of [46], at a loss of $(1 + \epsilon)$ factor we can assume we have a set $\mathcal{C}$ of "candidate" centres from which the centres can be chosen from. This set can be computed in time $O(n\epsilon^{-d} \log(1/\epsilon))$ and $|\mathcal{C}| = O(n\epsilon^{-d} \log(1/\epsilon))$. Therefore, we can reduce the problem to the discrete case.

Formally, suppose we are given a set $\mathcal{C}$ of points (as possible cluster centres) along with $\mathcal{X}$ and we have to select the k centres from $\mathcal{C}$. Furthermore, we assume the points are given in a metric space (V, δ) (not necessarily $\mathbb{R}^d$). For any two points $p, q \in V$, $\delta(p, q)$ denotes the distance between them: for the case of the metric being $\mathbb{R}^d$, then $\delta(p, q) = \sqrt{\sum_{\ell=1}^{d} |p_\ell - q_\ell|^2}$.

Algorithm 1 ρ-Swap Local Search

Let $\mathcal{S}$ be an arbitrary set of k centres from $\mathcal{C}$
while $\exists$ sets $P \subseteq \mathcal{C} - \mathcal{S}$, $Q \subseteq \mathcal{S}$ with $|P| = |Q| \leq \rho$ s.t. $\text{cost}((\mathcal{S} - Q) \cup P) < \text{cost}(\mathcal{S})$ **do**
  $\mathcal{S} \leftarrow (\mathcal{S} - Q) \cup P$
return $\mathcal{S}$

We usually refer to a potential centre in $\mathcal{C}$ by a simple index i and a point in $\mathcal{X}$ by a simple index j (or slight variants like i^* or $\overline{i'}$). This is to emphasize that we do not need to talk about specific coordinates of points in Euclidean space. In fact, only once in our proof do we rely on the particular embedding of the points in Euclidean space. This argument will also be replaced by a more general argument when discussing doubling metrics in Section 5.1. So, for any set $S \subseteq \mathcal{C}$ and any $j \in \mathcal{X}$, let $\delta(j, S) = \min_{i \in S} \delta(j, i)$. We also define $\text{cost}(S) = \sum_{j \in \mathcal{X}} \delta(j, S)^2$.

Our goal in (discrete) k-MEANS is to find a set of centres $S \subseteq \mathcal{C}$ of size k to minimize cost(S). Note that once we fix the set of centres we can find a partitioning of $\mathcal{X}$ that realizes $\sum_{j \in \mathcal{X}} \delta(j, S)^2$ by assigning each $j \in \mathcal{X}$ to the nearest centre in S, breaking ties arbitrarily.

For ease of exposition we focus on k-MEANS on $\mathbb{R}^d$ and then show how the analysis can be extended to work for metrics with constant doubling dimension and when we want to minimize $\sum_{j \in \mathcal{X}} \delta(j, S)^q$ for a fixed $q \geq 1$ instead of just $q = 2$.

The simple ρ-swap local search heuristic shown in Algorithm 1 is essentially the same one considered in [36].

Recall that we defined $\rho(\epsilon, d) = d^{O(d)} \cdot \epsilon^{O(d/\epsilon)}$, where the constants will be specified later and consider the local search algorithm with $\rho = \rho(\epsilon, d)$ swaps. By a standard argument (as in [7, 36]) one can show that replacing the condition of the while loop with $\text{cost}((\mathcal{S} - Q) \cup P) \leq (1 - \frac{\epsilon}{k}) \cdot \text{cost}(\mathcal{S})$, the algorithm terminates in polynomial time. Furthermore, if α is such that any locally optimum solution returned by Algorithm 1 has cost at most $\alpha \cdot \text{cost}(\mathcal{O})$ where $\mathcal{O}$ denotes a global optimum solution, then any $\mathcal{S}$ such that $\text{cost}((\mathcal{S} - Q) \cup P) < (1 - \frac{\epsilon}{k}) \cdot \text{cost}(\mathcal{S})$ for any possible swap P, Q satisfies $\text{cost}(\mathcal{S}) \leq \frac{\alpha}{1-\epsilon} \text{cost}(\mathcal{O})$. This follows by arguments in [7, 36] and the fact that our local search analysis uses at most k "test swaps".

For ease of exposition, we ignore this factor $1 + \epsilon$ loss, and consider the solution $\mathcal{S}$ returned by Algorithm 1. Recall that we use $\mathcal{O}$ to denote the global optimum solution. For $j \in \mathcal{X}$, let $c_j^* = \delta(j, \mathcal{O})^2$ and $c_j = \delta(j, \mathcal{S})^2$, so $\text{cost}(\mathcal{O}) = \sum_{j \in \mathcal{X}} c_j^*$ and $\text{cost}(\mathcal{S}) = \sum_{j \in \mathcal{X}} c_j$. We also denote the centre in $\mathcal{O}$ nearest to j by $\sigma^*(j)$ and the centre in $\mathcal{S}$ nearest to j by $\sigma(j)$. Define $\phi : \mathcal{O} \cup \mathcal{S} \to \mathcal{O} \cup \mathcal{S}$ to be the function that assigns $i^* \in \mathcal{O}$ to its nearest centre in $\mathcal{S}$ and assigns $i \in \mathcal{S}$ to its nearest centre in $\mathcal{O}$. For any two sets $S, T \subseteq \mathcal{O} \cup \mathcal{S}$, we let $S \triangle T = (S \cup T) - (S \cap T)$.

We assume $\mathcal{O} \cap \mathcal{S} = \emptyset$. This is without loss of generality because we could duplicate each location in $\mathcal{C}$ and say $\mathcal{O}$ uses the originals and $\mathcal{S}$ the duplicates. It is easy to check that $\mathcal{S}$ would still be a locally optimum solution in this instance. We can also assume that these are the only possible colocated facilities, so $\delta(i, i') > 0$ for distinct $i, i' \in \mathcal{O}$ or distinct $i, i' \in \mathcal{S}$. Finally, we will assume ϵ is sufficiently small (independent of all other parameters, including d) so that all of our bounds hold.

3 Local Search Analysis for $\mathbb{R}^d$

In this section we focus on $\mathbb{R}^d$ (for fixed $d \geq 2$) and define $\rho(\epsilon, d) = 32 \cdot (2d)^{8d} \cdot \epsilon^{-36 \cdot d/\epsilon}$. Our goal in this section is to prove that the ρ-swap local search with $\rho = \rho(\epsilon, d)$ is a PTAS for k-MEANS in $\mathbb{R}^d$.

Algorithm 2 Filtering $\mathcal{O}$

$\overline{\mathcal{O}} \leftarrow \emptyset$
for each $i^* \in \mathcal{O}$ in nondecreasing order of D_{i^*} **do**
  if $\exists\ \overline{i^*} \in \overline{\mathcal{O}}$ such that $\delta(i^*, \overline{i^*}) \leq \epsilon \cdot D_{i^*}$ **then**
    $\eta(i^*) \leftarrow \overline{i^*}$
  else
    $\eta(i^*) \leftarrow i^*$
    $\overline{\mathcal{O}} \leftarrow \overline{\mathcal{O}} \cup \{i^*\}$
return $\overline{\mathcal{O}}$

Theorem 4 *Let $\mathcal{S}$ be a locally-optimum solution with respect to the $\rho(\epsilon, d)$-swap local search heuristic when the points lie in $\mathbb{R}^d$. Then* $\text{cost}(\mathcal{S}) \leq (1 + O(\epsilon)) \cdot \text{cost}(\mathcal{O})$.

To prove this, we will construct a set of test swaps that yield various inequalities which, when combined, provide the desired bound on $\text{cost}(\mathcal{S})$. That is, we will partition $\mathcal{O} \cup \mathcal{S}$ into sets where $|P \cap \mathcal{O}| = |P \cap \mathcal{S}| \leq \rho(\epsilon, d)$ for each part P. For each such set P, $0 \leq \text{cost}(\mathcal{S} \triangle P) - \text{cost}(\mathcal{S})$ because $\mathcal{S}$ is a locally optimum solution. We will provide an explicit upper bound on this cost change that will reveal enough information to easily conclude $\text{cost}(\mathcal{S}) \leq (1 + O(\epsilon)) \cdot \text{cost}(\mathcal{O})$. For example, for a point $j \in \mathcal{X}$ if $\sigma^*(j) \in P$ then the change in j's assignment cost is at most $c_j^* - c_j$ because we could assign j from $\sigma(j)$ to $\sigma^*(j)$. The problem is that points j with $\sigma(j) \in P$ but $\sigma^*(j) \notin P$ must go somewhere else; most of our effort is ensuring that the test swaps are carefully chosen so such reassignment cost increases are very small.

First we need to describe the partition of $\mathcal{O} \cup \mathcal{S}$. This is a fairly elaborate scheme that involves several steps. As mentioned earlier, the actual algorithm for k-MEANS is the simple local search we described and the algorithms we describe below to get this partitioning scheme are only for the purpose of proof and analysis of the local search algorithm.

Definition 1 *For $i^* \in \mathcal{O}$ let $D_{i^*} := \delta(i^*, \mathcal{S}) = \delta(i^*, \phi(i^*))$. For $i \in \mathcal{S}$ let $D_i := \delta(i, \mathcal{O}) = \delta(i, \phi(i))$.*

The first thing is to sparsify $\mathcal{O}$ and $\mathcal{S}$ using a simple filtering step. Algorithm 2 *filters* $\mathcal{O}$ to a set that is appropriately sparse for our analysis.

Think of $\eta(i^*)$ as a proxy for i^* that is very close to i^*. Using a similar process, we filter $\mathcal{S}$ to get $\overline{\mathcal{S}}$ and proxy centres $\eta(i) \in \overline{\mathcal{S}}$ for each $i \in \mathcal{S}$. The idea is that the set of centres left in $\overline{\mathcal{O}}$ and $\overline{\mathcal{S}}$ are somewhat far apart yet any point that was assigned to a centre in $\mathcal{O} - \overline{\mathcal{O}}$ (or in $\mathcal{S} - \overline{\mathcal{S}}$) can be "cheaply" reassigned to a proxy.

Lemma 1 *For each $i \in \mathcal{O} \cup \mathcal{S}$ we have $\delta(i, \eta(i)) \leq \epsilon \cdot D_i$. For any distinct $i, i' \in \overline{\mathcal{O}} \cup \overline{\mathcal{S}}$ we have $\delta(i, i') \geq \epsilon \cdot \max\{D_i, D_{i'}\}$.*

Proof. That $\delta(i, \eta(i)) \leq \epsilon \cdot D_i$ follows immediately by construction. If $i \in \overline{\mathcal{O}}, i' \in \overline{\mathcal{S}}$ or vice-versa, then in fact $\delta(i, i') \geq \max\{D_i, D_{i'}\}$ simply by definition of $D_i, D_{i'}$.

Now suppose $i, i' \in \overline{\mathcal{O}}$ and that i' was considered after i in Algorithm 2 (so $D_{i'} \geq D_i$). The fact that i' was added to $\overline{\mathcal{O}}$ even though i was already in $\overline{\mathcal{O}}$ means $\delta(i, i') \geq \epsilon \cdot D_{i'}$. The same argument works if $i, i' \in \overline{\mathcal{S}}$. ∎

Next we define mappings similar to ϕ, σ, σ^* except they only concern centres that were not filtered out.

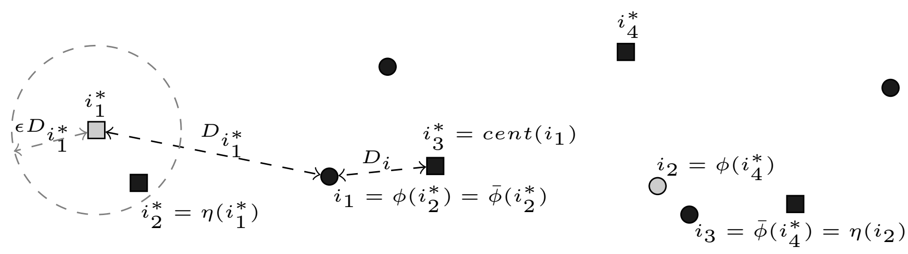

Figure 1: The square centres lie in $\mathcal{O}$ and the circle centres lie in $\mathcal{S}$. The black ones survived the filtering (i.e. lie in $\overline{\mathcal{O}} \cup \overline{\mathcal{S}}$) and the grey ones were filtered out. Note that the grey square i_1^* on the left was too close to i_2^* so it was not added to $\overline{\mathcal{O}}$ and $\eta(i_1^*) = i_2^*$. Also note that $\overline{\phi}(i_4^*) \neq \phi(i_4^*)$ because the facility i_2 that defined $D_{i_4^*} := \delta(i_4^*, i_2)$ was not added to $\overline{\mathcal{S}}$. Still, $\delta(i_4^*, \overline{\phi}(i_4^*)) = \delta(i_4^*, i_3) \leq (1+\epsilon) D_{i_4^*}$.

Definition 2

- $\overline{\phi} : \overline{\mathcal{O}} \cup \overline{\mathcal{S}} \to \overline{\mathcal{O}} \cup \overline{\mathcal{S}}$ *maps each* $i \in \overline{\mathcal{O}}$ *to its nearest location in* $\overline{\mathcal{S}}$ *and vice versa.*
- $\overline{\sigma}^* : \mathcal{X} \to \overline{\mathcal{O}}$ *defined by* $\overline{\sigma}^*(j) = \eta(\sigma^*(j))$.
- $\overline{\sigma} : \mathcal{X} \to \overline{\mathcal{S}}$ *defined by* $\overline{\sigma}(j) = \eta(\sigma(j))$.

Finally, for each $i \in \overline{\phi}(\overline{\mathcal{O}})$, *let* cent$(i)$ *be the centre in* $\overline{\phi}^{-1}(i)$ *that is closest to* i, *breaking ties arbitrarily.*

Note $\overline{\sigma}(j)$ may not necessarily be the centre in $\overline{\mathcal{S}}$ that is closest to j. Also note that if one considers a bipartite graph with parts $\overline{\mathcal{O}}$ and $\overline{\mathcal{S}}$, then $\overline{\phi}$ maps centres from one side to the other.

Lemma 2 *For each* $i' \in \overline{\mathcal{O}} \cup \overline{\mathcal{S}}$, $D_{i'} \leq \delta(i', \overline{\phi}(i')) \leq (1+\epsilon) \cdot D_{i'}$.

Proof. Suppose $i' \in \overline{\mathcal{O}}$, the proof is essentially the same for $i' \in \overline{\mathcal{S}}$. On one hand, we know

$$D_{i'} = \delta(i', \phi(i')) \leq \delta(i', \overline{\phi}(i'))$$

because $\overline{\mathcal{S}} \subseteq \mathcal{S}$. On the other hand,

$$\delta(i', \overline{\phi}(i')) = \delta(i', \eta(\phi(i'))) \leq \delta(i', \phi(i')) + \delta(\phi(i'), \eta(\phi(i'))) \leq D_{i'} + \epsilon \cdot D_{\phi(i')}.$$

Conclude by observing $D_{\phi(i')} \leq \delta(i', \phi(i')) = D_{i'}$. ■

Figure 1 depicts many of the concepts covered above. Finally, the last definition in this section identifies pairs of centres that we would like to have in the same part of the partition we construct.

Definition 3

- $\mathcal{T} := \{(\text{cent}(i), i) : i \in \overline{\phi}(\overline{\mathcal{O}}) \text{ and } \epsilon \cdot \delta(\text{cent}(i), i) \leq D_i\}$
- $\mathcal{N} := \{(i^*, i) \in \overline{\mathcal{O}} \times \overline{\mathcal{S}} : \delta(i, i^*) \leq \epsilon^{-1} \cdot D_i \text{ and } D_{i^*} \geq \epsilon \cdot D_i\}$

For each $i \in \overline{\mathcal{S}}$, the set $\{i^* : (i^*, i) \in \mathcal{N}\}$ is the "net" for centre i that was discussed in the proof outline in Section 1.2.

Ultimately we will require that pairs in $\mathcal{T}$ are not separated by the partition. Our requirement for $\mathcal{N}$ is not quite as strong. The partition is constructed randomly and it will be sufficient to have each pair in $\mathcal{N}$ being separated by the partition with probability at most ϵ.

3.1 Every Centre is Close to Some Pair in $\mathcal{T}$

The following says that if at least one centre of each pair in $\mathcal{T}$ is open after a swap, then every centre in $\overline{\mathcal{O}} \cup \overline{\mathcal{S}}$ is somewhat close to some open centre. The bound is a bit big, but it will be multiplied by $O(\epsilon)$ whenever it is used in the local search analysis.

Lemma 3 *Let $A \subseteq \overline{\mathcal{O}} \cup \overline{\mathcal{S}}$ be such that $A \cap \{\mathrm{cent}(i), i\} \neq \emptyset$ for each $(\mathrm{cent}(i), i) \in \mathcal{T}$. Then $\delta(i', A) \leq 5 \cdot D_{i'}$ for any $i' \in \mathcal{O} \cup \mathcal{S}$.*

Proof. We first prove the statement for $i' \in \mathcal{O}$, the other case is similar but requires one additional step so we will discuss it below. Consider the following sequence of centres. Initially, set $i_0 := i'$, $i_1 := \eta(i_0)$, and $i_2 := \overline{\phi}(i_1)$. We build the rest inductively, noting that we guarantee $i_a \in \overline{\phi}(\overline{\mathcal{O}})$ for even indices $a \geq 2$ (so $\mathrm{cent}(i_a)$ is defined).

Inductively, for even $a \geq 2$ we do the following. If $i_a \in A$ then we stop. Otherwise, if $(\mathrm{cent}(i_a), i_a) \in \mathcal{T}$ then by assumption it must be that $\mathrm{cent}(i_a) \in A$ so we let $i_{a+1} := \mathrm{cent}(i_a)$ and stop. Finally, if $(\mathrm{cent}(i_a), i_a) \notin \mathcal{T}$ then we set $i_{a+1} := \overline{\phi}(i_a)$ and $i_{a+2} := \overline{\phi}(i_{a+1})$ and iterate with $a' = a + 2$. This walk is depicted in Figure 2.

We will soon show the walk terminates. For now, we observe that, apart from the first step, the steps decrease in length geometrically. In particular, consider some $a \geq 2$ such that the walk did not stop at i_a. If $i_{a+1} = \overline{\phi}(i_a)$ then

$$\delta(i_a, i_{a+1}) = \delta(i_a, \overline{\phi}(i_a)) \leq \delta(i_a, i_{a-1}).$$

If $i_{a+1} \neq \overline{\phi}(i_a)$ then it must be $i_a = \mathrm{cent}(i_{a-1})$. In this case, it must be $i_a = \overline{\phi}(i_{a-1})$, so because $\mathrm{cent}(i_a)$ is the closest centre in $\overline{\phi}^{-1}(i_a)$ to i_a (by definition), we have

$$\delta(i_a, i_{a+1}) = \delta(i_a, \mathrm{cent}(i_a)) \leq \delta(i_a, i_{a-1}).$$

We prove that the lengths of the edges traversed decrease geometrically with every other step. This, along with the fact that the lengths of the steps of the walk are nonincreasing (except, perhaps, the first two steps), will show that this process eventually terminates and also bounds the cost of the path.

Claim 1 *For every even $a \geq 2$ such that the walk did not end at i_a or i_{a+1}, we have $\delta(i_a, i_{a+1}) \leq 2\epsilon \cdot \delta(i_{a-1}, i_a)$.*

Proof. Because the walk did not end at i_a or i_{a+1}, $(\mathrm{cent}(i_a), i_a) \notin \mathcal{T}$ meaning $D_{i_a} < \epsilon \cdot \delta(\mathrm{cent}(i_a), i_a) \leq \epsilon \cdot \delta(i_{a-1}, i_a)$. Using this and Lemma 2 in the first bound below, we see

$$\delta(i_a, i_{a+1}) \leq (1+\epsilon) \cdot D_{i_a} < \epsilon(1+\epsilon) \cdot \delta(i_{a-1}, i_a) \leq 2\epsilon \cdot \delta(i_{a-1}, i_a)$$

■

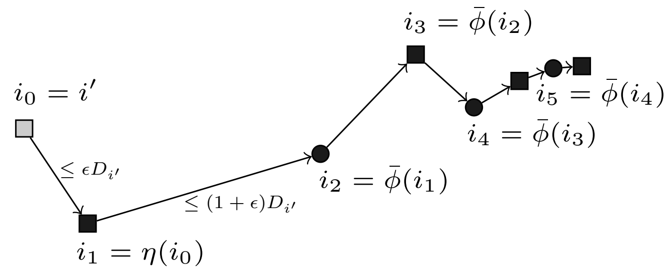

Figure 2: Illustration of the walk $i_0, i_1, \ldots$ in the proof of Lemma 3. Apart from $a = 1$, the step (i_a, i_{a+1}) is shorter than the step (i_{a-1}, i_a). Furthermore, every other step decreases in length geometrically.

Let m be the index of the last centre in the walk. Thus, $\delta(i_a, i_{a+1}) \leq \delta(i_{a-1}, i_a)$ for all $2 \leq a \leq m-1$. From this and Claim 1 we have

$$\begin{aligned}
\delta(i', A) &\leq \sum_{j=0}^{m-1} \delta(i_j, i_{j+1}) \\
&\leq \delta(i_0, i_1) + \sum_{\substack{2 \leq a \leq m \\ a \text{ even}}} 2\delta(i_{a-1}, i_a) + \delta(i_{m-1}, i_m) \\
&\leq \delta(i_0, i_1) + 2\delta(i_1, i_2) \sum_{a \geq 0} (2\epsilon)^a + \delta(i_{m-1}, i_m) \\
&\leq \delta(i_0, i_1) + \frac{2}{1-2\epsilon} \cdot \delta(i_1, i_2) + \delta(i_1, i_2) \\
&= \delta(i_0, i_1) + \frac{3-2\epsilon}{1-2\epsilon} \cdot \delta(i_1, i_2) \\
&\leq \epsilon \cdot D_{i'} + \frac{(3-2\epsilon)(1+\epsilon)}{1-2\epsilon} \cdot D_{\eta(i')} \\
&\leq (3+7\epsilon) \cdot D_{i'}.
\end{aligned}$$

The second last step uses Lemma 2 and the last step uses the fact that $D_{\eta(i')} \leq D_{i'}$ (either $i' = \eta(i')$ or else i' was filtered out by $\eta(i')$, in which case it has a larger D-value) and the assumption that ϵ is small enough.

Now suppose $i' \in \mathcal{S}$. We bound $\delta(i', A)$ mostly using what we have done already. That is, we have

$$\delta(i', A) \leq \delta(i', \phi(i')) + \delta(\phi(i'), A) \leq D_{i'} + (3+7\epsilon) D_{\phi(i')}.$$

Note $D_{\phi(i')} \leq \delta(i', \phi(i')) = D_{i'}$ again by Lemma 2. So,

$$\delta(i', A) \leq (4+7\epsilon) D_{i'} \leq 5 D_{i'}.$$

■

3.2 Good Partitioning of $\mathcal{O} \cup \mathcal{S}$ and Proof of Theorem 4

The main tool used in our analysis is the existence of the following randomized partitioning scheme.

Theorem 5 *There is a randomized algorithm that samples a partitioning π of $\mathcal{O} \cup \mathcal{S}$ such that:*

- *For each part $P \in \pi$, $|P \cap \mathcal{O}| = |P \cap \mathcal{S}| \leq \rho(\epsilon, d)$.*
- *For each part $P \in \pi$, $\mathcal{S} \triangle P$ includes at least one centre from every pair in $\mathcal{T}$.*
- *For each $(i^*, i) \in \mathcal{N}$, $\Pr[i, i^* \text{ lie in different parts of } \pi] \leq \epsilon$.*

We prove this theorem in Section 4. For now, we will complete the analysis of the local search algorithm using this partitioning scheme. Note that in the following we do not use the geometry of the metric (i.e. all arguments hold for general metrics); it is only in the proof of Theorem 5 that we use properties of $\mathbb{R}^d$.

The following gives a way to handle the fact that the triangle inequality does not hold with squares of the distances.

Lemma 4 *For any real numbers x, y we have $(x+y)^2 \leq 2(x^2+y^2)$.*

Proof. $(x+y)^2 \leq (x+y)^2 + (x-y)^2 = 2x^2 + 2y^2$. ■

Lemma 5 *For each point $j \in \mathcal{X}$, $D_{\overline{\sigma}(j)} \leq D_{\sigma(j)} \leq \delta(j, \sigma(j)) + \delta(j, \sigma^*(j))$. Similarly, $D_{\overline{\sigma}^*(j)} \leq D_{\sigma^*(j)} \leq \delta(j, \sigma(j)) + \delta(j, \sigma^*(j))$.*

Proof. As usual, we only prove the first statement since the second is nearly identical. If $\overline{\sigma}(j) = \sigma(j)$ then $D_{\overline{\sigma}(j)} = D_{\sigma(j)}$ is trivially true. Otherwise, $\overline{\sigma}(j) = \eta(\sigma(j))$ was already in $\overline{\mathcal{S}}$ when $\sigma(j)$ was considered by the filtering algorithm meaning $D_{\overline{\sigma}(j)} \leq D_{\sigma(j)}$.

For the other inequality, note

$$D_{\sigma(j)} = \delta(\sigma(j), \phi(\sigma(j))) \leq \delta(\sigma(j), \sigma^*(j)) \leq \delta(j, \sigma(j)) + \delta(j, \sigma^*(j)).$$

■

Proof of Theorem 4.
Let π be a partition sampled by the algorithm from Theorem 5. For each point $j \in \mathcal{X}$ and each part P of π, let $\Delta_j^P := \delta(j, \mathcal{S} \triangle P)^2 - \delta(j, \mathcal{S})^2$ denote the change in assignment cost for the point after swapping in the centers in $P \cap \mathcal{O}$ and swapping out $P \cap \mathcal{S}$. Local optimality of $\mathcal{S}$ and $|P \cap \mathcal{S}| = |P \cap \mathcal{O}| \leq \rho(\epsilon, d)$ means $0 \leq \sum_j \Delta_j^P$ for any part P.

Classify each point $j \in \mathcal{X}$ in one of the following ways:

- **Lucky**: $\sigma(j)$ and $\overline{\sigma}(j)$ do not lie in the same part of π.
- **Long**: j is not lucky but $\delta(\overline{\sigma}(j), \overline{\sigma}^*(j)) > \epsilon^{-1} \cdot D_{\overline{\sigma}(j)}$.
- **Bad**: j is not lucky or long and $(\overline{\sigma}^*(j), \overline{\sigma}(j)) \in \mathcal{N}$ yet $\overline{\sigma}(j), \overline{\sigma}^*(j)$ lie in different parts of π.
- **Good**: j is neither lucky, long, nor bad.

We now place an upper bound on $\sum_{P \in \pi} \Delta_j^P$ for each point $j \in \mathcal{X}$. Note that each centre in $\mathcal{S}$ is swapped out exactly once over all swaps P and each centre in $\mathcal{O}$ is swapped in exactly once. With this in mind, consider the following cases for a point $j \in \mathcal{X}$. In the coming arguments, we let $\delta_j := \delta(j, \sigma(j))$ and $\delta_j^* := \delta(j, \sigma^*(j))$ for brevity. Note $c_j = \delta_j^2$ and $c_j^* = \delta_j^{*2}$.

In all cases for j except when j is bad, the main idea is that we can bound the distance from j to some point in $\mathcal{S} \triangle P$ by first moving it to either $\sigma(j)$ or $\sigma^*(j)$ and then moving it a distance of $O(\epsilon) \cdot (\delta_j + \delta_j^*)$ to reach an open facility. Considering that we reassigned j from $\sigma(j)$, the reassignment cost will be

$$(\delta_j + O(\epsilon) \cdot (\delta_j + \delta_j^*))^2 - c_j = O(\epsilon) \cdot (c_j + c_j^*)$$

or

$$(\delta_j^* + O(\epsilon) \cdot (\delta_j + \delta_j^*))^2 - c_j = (1 + O(\epsilon)) \cdot c_j^* - (1 - O(\epsilon)) \cdot c_j.$$

Case: j is lucky
For the part $P \in \pi$ with $\sigma^*(j) \in P$, we have $\Delta_j^P \leq c_j^* - c_j$ as we could move j from $\sigma(j)$ to $\sigma^*(j)$. If $\sigma(j)$ is swapped out in a different swap P', we move j to $\overline{\sigma}(j)$ (which remains open because j is lucky) and bound $\Delta_j^{P'}$ by:

$$\begin{aligned}
\Delta_j^{P'} &\leq \delta(j, \overline{\sigma}(j))^2 - \delta_j^2 \\
&\leq (\delta_j + \delta(\sigma(j), \overline{\sigma}(j)))^2 - c_j \\
&\leq (\delta_j + \epsilon \cdot D_{\sigma(j)})^2 - c_j && \text{(Lemma 1)} \\
&= 2\epsilon \cdot \delta_j \cdot D_{\sigma(j)} + \epsilon^2 \cdot D_{\sigma(j)}^2 \\
&\leq 2\epsilon \cdot \delta_j \cdot (\delta_j + \delta_j^*) + \epsilon^2 \cdot (\delta_j + \delta_j^*)^2 && \text{(Lemma 5)} \\
&\leq 2\epsilon \cdot (\delta_j + \delta_j^*)^2 + \epsilon^2 \cdot (\delta_j + \delta_j^*)^2 \\
&\leq 4\epsilon \cdot (c_j^* + c_j) + 2\epsilon^2 \cdot (c_j^* + c_j) && \text{(Lemma 4)} \\
&\leq 6\epsilon \cdot (c_j^* + c_j),
\end{aligned}$$

again using the assumption that ϵ is sufficiently small. For every other swap P'', we have that $\sigma(j)$ remains open after the swap so $\Delta_j^{P''} \leq 0$ as we could just leave j at $\sigma(j)$. In total, we have

$$\sum_{P \in \pi} \Delta_j^P \leq c_j^* - c_j + 6\epsilon \cdot (c_j^* + c_j).$$

Case: j is long
Again, for $P \in \pi$ with $\sigma^*(j) \in P$ we get $\Delta_j^P \leq c_j^* - c_j$. If $\sigma(j)$ is swapped out in a different swap P', then we bound $\Delta_j^{P'}$ by moving j from $\sigma(j)$ to the open centre nearest to $\sigma(j)$. Note that $\mathcal{S} \triangle P'$ contains at least one centre from every pair in $\mathcal{T}$, so we bound this distance using Lemma 3. This case is depicted in Figure 3. We have

$$\begin{aligned}
D_{\sigma(j)} &\leq \delta(\sigma(j), \overline{\sigma}(j)) + \delta(\overline{\sigma}(j), \phi(\overline{\sigma}(j))) \\
&\leq \epsilon D_{\sigma(j)} + D_{\overline{\sigma}(j)} && \text{(Lemma 1)} \\
&\leq \epsilon(\delta(j, \sigma(j)) + \delta(j, \sigma^*(j)) + \epsilon\delta(\overline{\sigma}(j), \overline{\sigma}^*(j)) && \text{(since } j \text{ is long)} \\
&\leq \epsilon(\delta_j + \delta_j^*) + \epsilon(\delta(\overline{\sigma}(j), \sigma(j)) + \delta(\sigma(j), j) + \delta(j, \sigma^*(j)) + \delta(\sigma^*(j), \overline{\sigma}^*(j))) \\
&\leq \epsilon(\delta_j + \delta_j^*) + \epsilon(\epsilon D_{\sigma(j)} + \delta_j + \delta_j^* + \epsilon D_{\sigma^*(j)}) && \text{(Lemma 1)} \\
&\leq \epsilon(\delta_j + \delta_j^*) + \epsilon(\epsilon(\delta_j + \delta_j^*) + \delta_j + \delta_j^* + \epsilon(\delta_j + \delta_j^*)) && \text{(Lemma 5)} \\
&= 2\epsilon(1 + \epsilon)(\delta_j + \delta_j^*).
\end{aligned}$$

Using this, we bound $\Delta_j^{P'}$ as follows.

$$\begin{aligned}
\Delta_j^{P'} &\leq (\delta_j + \delta(\sigma(j), \mathcal{S}\triangle P))^2 - c_j \\
&\leq (\delta_j + 5D_{\sigma(j)})^2 - c_j && \text{(Lemma 3)} \\
&\leq (\delta_j + 10\epsilon(1+\epsilon)\cdot(\delta_j + \delta_j^*))^2 - c_j && \text{bound for } D_{\sigma(j)} \text{ above} \\
&\leq 21\epsilon\cdot\delta_j(\delta_j+\delta_j^*) + 101\epsilon^2\cdot(\delta_j+\delta_j^*)^2 \\
&\leq 22\epsilon\cdot(\delta_j+\delta_j^*)^2 \\
&\leq 44\epsilon\cdot(c_j+c_j^*).
\end{aligned}$$

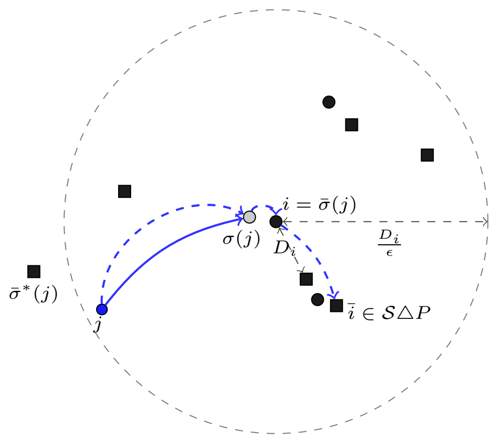

Figure 3: Illustrating the case when j is long. It is moved from $\sigma(j)$ to the nearest open centre at an additional distance of $O(D_{\sigma(j)})$. This is negligible compared to $\delta_j + \delta_j^*$.

In every other swap we could leave j at $\sigma(j)$. Thus, for a long point j we have

$$\sum_{P\in\pi} \Delta_j^P \leq c_j^* - c_j + 44\epsilon\cdot(c_j^*+c_j).$$

Case: j is bad

We only move j in the swap P when $\sigma(j)$ is closed. In this case, we assign j in the same way as if it was long. Our bound is weaker here and introduces significant positive dependence on c_j. This will eventually be compensated by the fact that j is bad with probability at most ϵ over the random choice of π. For now, we just provide the reassignment cost bound for bad j.

$$\begin{aligned}
\Delta_j^{P} &\leq (\delta_j + \delta(\sigma(j), \mathcal{S}\triangle P))^2 - c_j \\
&\leq (\delta_j + 5D_{\sigma(j)})^2 - c_j && \text{(Lemma 3)} \\
&\leq (6\delta_j + 5\delta_j^*)^2 - c_j && \text{(Lemma 4)} \\
&\leq 71\cdot(c_j^*+c_j).
\end{aligned}$$

So for bad points we have

$$\sum_{P\in\pi} \Delta_j^P \leq 71\cdot(c_j^*+c_j).$$

Case: j is good
This breaks into two subcases. We know $\delta(\overline{\sigma}^*(j), \overline{\sigma}(j)) \leq \epsilon^{-1} \cdot D_{\overline{\sigma}(j)}$ because j is not long. In one subcase, $D_{\overline{\sigma}^*(j)} \geq \epsilon \cdot D_{\overline{\sigma}(j)}$ so $(\overline{\sigma}^*(j), \overline{\sigma}(j)) \in \mathcal{N}$. Since j is not bad and not lucky, we have $\sigma(j), \overline{\sigma}(j), \overline{\sigma}^*(j) \in P$ for some common part $P \in \pi$. In the other subcase, $D_{\overline{\sigma}^*(j)} < \epsilon \cdot D_{\overline{\sigma}(j)}$. Note in this case we still have $\sigma(j), \overline{\sigma}(j) \in P$ for some common part P because j is not lucky.

Subcase: $D_{\overline{\sigma}^*(j)} \geq \epsilon \cdot D_{\overline{\sigma}(j)}$
The only time we move j is when $\sigma(j)$ is closed. As observed in the previous paragraph, this happens in the same swap when $\overline{\sigma}^*(j)$ is opened, so send j to $\overline{\sigma}^*(j)$. This is illustrated in Figure 4.

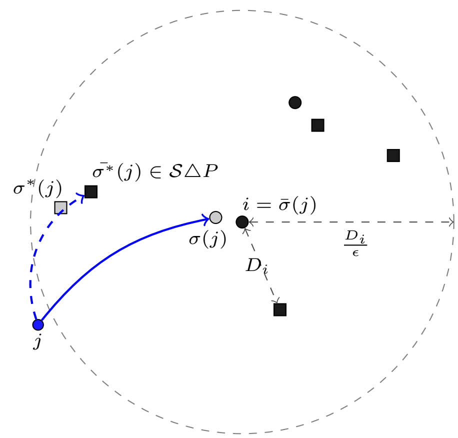

Figure 4: Illustrating the subcase when $D_{\overline{\sigma}^*(j))} \geq \epsilon \cdot D_{\overline{\sigma}(j)}$. In this case, the only additional distance j travels after first being reassigned to $\sigma^*(j)$ is to the nearby proxy $\overline{\sigma}^*(j)$

$$
\begin{aligned}
\Delta_j^P &\leq (\delta_j^* + \delta(\sigma^*(j), \overline{\sigma}^*(j)))^2 - c_j \\
&\leq (\delta_j^* + \epsilon \cdot D_{\sigma^*(j)})^2 - c_j && \text{(Lemma 1)} \\
&\leq (\delta_j^* + \epsilon \cdot (\delta_j + \delta_j^*))^2 - c_j \\
&= c_j^* + \epsilon^2 \cdot (\delta_j + \delta_j^*)^2 + 2\epsilon \cdot \delta_j^*(\delta_j + \delta_j^*) - c_j \\
&\leq c_j^* + 2\epsilon^2 \cdot (c_j + c_j^*) + 2\epsilon \cdot (\delta_j + \delta_j^*)^2 - c_j && \text{(Lemma 4)} \\
&\leq c_j^* - c_j + 5\epsilon \cdot (c_j^* + c_j). && \text{(Lemma 4)}
\end{aligned}
$$

Subcase: $D_{\overline{\sigma}^*(j)} < \epsilon \cdot D_{\overline{\sigma}(j)}$
Again, the only time we move j is when $\sigma(j)$ is closed. We reassign j by first moving it to $\sigma^*(j)$ and then using Lemma 3 to further bound the cost. Figure 5 depicts this reassignment.

We bound the cost change for this reassignment as follows. Recall that $\delta(\sigma^*(j), \overline{\sigma}^*(j)) \leq \epsilon D_{\sigma^*(j)}$ by our filtering. This implies that $D_{\sigma^*(j)} \leq D_{\overline{\sigma}^*(j)} + \epsilon D_{\sigma^*(j)}$ which in turn implies $D_{\sigma^*(j)} \leq \frac{1}{1-\epsilon} D_{\overline{\sigma}^*(j)}$; thus $D_{\sigma^*(j)} \leq (1 + 2\epsilon) D_{\overline{\sigma}^*(j)}$ which in turn is bounded by $\epsilon(1 + \epsilon) D_{\overline{\sigma}(j)}$ in this subcase

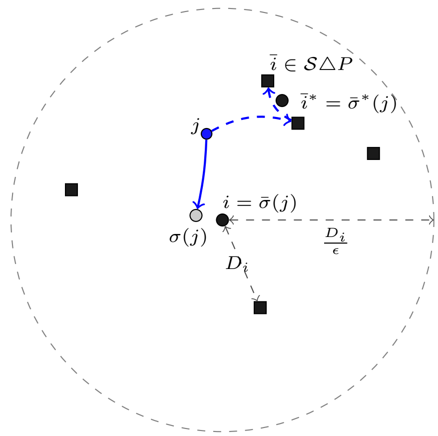

Figure 5: Illustrating the subcase when $D_{\overline{\sigma}^*(j)} < \epsilon \cdot D_{\overline{\sigma}(j)}$. The only additional distance j moves after being reassigned to $\sigma^*(j)$ is small, as it travels first to the nearby proxy $\overline{\sigma}^*(j)$ and then at distance at most $5D_{\overline{\sigma}^*(j)} \leq 5\epsilon \cdot D_{\overline{\sigma}(j)}$ by Lemma 3.

(by the assumption of subcase). Thus,

$$\begin{aligned}
\Delta_j^P &\leq (\delta_j^* + \delta(\sigma^*(j), \mathcal{S}\triangle P))^2 - c_j \\
&\leq (\delta_j^* + 5D_{\sigma^*(j)}) - c_j \\
&\leq (\delta_j^* + 5\epsilon(1+2\epsilon) \cdot D_{\overline{\sigma}(j)})^2 - c_j \\
&\leq (\delta_j^* + 5\epsilon(1+2\epsilon) \cdot (\delta_j + \delta_j^*))^2 - c_j \qquad \text{(Lemma 5)} \\
&\leq c_j^* + 11\epsilon \cdot \delta_j^*(\delta_j + \delta_j^*) + 26\epsilon^2(\delta_j + \delta_j^*)^2 - c_j \\
&\leq c_j^* - c_j + 12\epsilon \cdot (\delta_j + \delta_j^*)^2 \\
&\leq c_j^* - c_j + 24\epsilon \cdot (c_j^* + c_j).
\end{aligned}$$

Considering both subcases we can say that for any good point j that

$$\sum_{P \in \pi} \Delta_j^P \leq c_j^* - c_j + 24\epsilon \cdot (c_j^* + c_j).$$

Aggregating these bounds and remembering $0 \leq \sum_{j \in \mathcal{X}} \Delta_j^P$ for each $P \in \pi$ because $\mathcal{S}$ is a locally optimum solution, we have

$$0 \leq \sum_{j \in \mathcal{X}} \sum_{P \in \pi} \Delta_j^P \leq \sum_{\substack{j \in \mathcal{X} \\ j \text{ not bad}}} \left[(1+44\epsilon)c_j^* - (1-44\epsilon)c_j\right] + \sum_{\substack{j \in \mathcal{X} \\ j \text{ bad}}} 71(c_j^* + c_j).$$

The last step is to average this inequality over the random choice of π. Note that any point $j \in \mathcal{X}$ is bad with probability at most ϵ by the guarantee in Theorem 5 and the definition of *bad*.

Thus, we see

$$
\begin{aligned}
0 &\leq \mathbf{E}_{\pi}\left[\sum_{j\in\mathcal{X}}\sum_{P\in\pi}\Delta_j^P\right] \\
&\leq \sum_{j\in\mathcal{X}} \Pr[j \text{ not bad}]\cdot\left[(1+44\epsilon)c_j^* - (1-44\epsilon)c_j\right] + \Pr[j \text{ bad}]\cdot 71(c_j^*+c_j) \\
&\leq \sum_{j\in\mathcal{X}} (1+44\epsilon)c_j^* - (1-\epsilon)\cdot(1-44\epsilon)c_j + 71\epsilon\cdot(c_j^*+c_j) \\
&\leq \sum_{j\in\mathcal{X}} (1+115\epsilon)c_j^* - (1-116\epsilon)c_j.
\end{aligned}
$$

Rearranging shows

$$\text{cost}(\mathcal{S}) \leq \frac{1+115\epsilon}{1-116\epsilon}\cdot\text{cost}(\mathcal{O}) \leq (1+O(\epsilon))\cdot\text{cost}(\mathcal{O}).$$

■

3.3 Running Time Analysis

Recall that we can go to the discrete case by finding a set $\mathcal{C}$ of size $O(n\epsilon^{-d}\log(1/\epsilon)$. The analysis of Arya et al. [7, 8] shows that the number of local search steps is at most $\log(cost(S_0)/cost(\mathcal{O}))/\log\frac{1}{1-\epsilon/k}$, where S_0 is the initial solution. This is polynomial in the total bit complexity of the input (i.e. the input size). So we focus on bounding the time complexity of each local search step. Since the number of swaps in each step is bounded by $\rho = \rho(\epsilon, d)$, a crude upper bound on the time complexity of each step is $O((n\epsilon^{-d}\log(1/\epsilon))^{\rho})$.

We can speed up this algorithm by using the idea of coresets. First observe that our local search algorithm extends to the weighted setting where each point $j \in \mathcal{X}$ has a weight $w(j)$ and the cost of a clustering with centres C is $\sum_{j\in\mathcal{X}} w(j)\cdot\delta^2(j,C)$.

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a weighted set of points with weight function $w : \mathcal{X} \to \mathbb{R}$. For any set $C \subseteq \mathbb{R}^d$ of size k we use $\mu_C(\mathcal{X})$ to denote $\sum_{j\in\mathcal{X}} w(x)\delta(j,C)^2$. The cost of the optimum k-MEANS solution for $\mathcal{X}$ is $\mu_{\mathcal{O}}(\mathcal{X},k) = \min_{C\subseteq\mathbb{R}^d:|C|=k}\mu_C(\mathcal{X})$.

Definition 4 *Let $\mathcal{X} \subseteq \mathbb{R}^d$ and $S \subseteq \mathbb{R}^d$ be two weighted point sets. For any k, ϵ, we say S is a (k,ϵ)-coreset if for every set of k centres $C \subseteq \mathbb{R}^d$:*

$$(1-\epsilon)\mu_C(\mathcal{X}) \leq \mu_C(S) \leq (1+\epsilon)\mu_C(\mathcal{X}).$$

A set $Z \subseteq \mathbb{R}^d$ is a (k,ϵ)-centroid set for $\mathcal{X}$ if there is a set $C \subseteq Z$ of size k such that $\mu_C(\mathcal{X}) \leq (1+\epsilon)\mu_{\mathcal{O}}(\mathcal{X},k)$.

Earlier works on coresets for k-means [30, 25, 26, 41] imply the existence of a (k,ϵ)-coresets of small size. For example, [30] show existence of (k,ϵ)-coresets of size $O(k^3/\epsilon^{d+1})$. Applying the result of [46], we get a (k,ϵ)-centroid set of size $O(\log(1/\epsilon)k^3/\epsilon^{2d+1})$ over a (k,ϵ)-coreset of size $O(k^3/\epsilon^{d+1})$. Thus running our local search ρ-swap algorithm takes $O((k/\epsilon)^{\zeta})$ time per iteration where $\zeta = d^{O(d)}\cdot\epsilon^{-O(d/\epsilon)}$.

4 The Partitioning Scheme: Proof of Theorem 5

Here is the overview of our partitioning scheme. First we bucket the real values by defining bucket a to have real values r where $\epsilon^{-a} \le r < \epsilon^{-(a+1)}$. Fix $b = \Theta(1/\epsilon)$ and then choose a random off-set a' and consider b consecutive buckets $a' + \ell \cdot b, \ldots, a' + (\ell+1) \cdot b - 1$ to form "bands". We get a random partition of points $i \in \overline{\mathcal{O}} \cup \overline{\mathcal{S}}$ into bands $B_1, B_2, \ldots$ such that all centres i in the same band B_ℓ have the property that their D_i values are in buckets $a' + \ell \cdot b, \ldots, a' + (\ell+1) \cdot b - 1$ (so $\epsilon^{-(a'+\ell \cdot b)} \le D_i < \epsilon^{-(a'+(\ell+1)\cdot b-1)}$). Given that for each pair $(i, i^*) \in \mathcal{N}$ the D_i, D_{i^*} values are within a factor $1/\epsilon$ of each other, the probability that they fall into different bands is small (like $\epsilon/4$). Next we impose a random grid partition on each band B_ℓ to cut it into cells with cell width roughly $\Theta(d/\epsilon^{\ell \cdot b})$; again the randomness comes from choosing a random off-set for our grid. The randomness helps to bound the probability of any pair $(i, i^*) \in \mathcal{N}$ being cut into two different cells to be small again. This will ensure the 3rd property of the theorem holds. Furthermore, since each $i \in B_\ell$ has $D_i \ge \epsilon^{-(a'+\ell \cdot b)}$ and by the filtering we did (Lemma 1), balls of radius $\frac{\epsilon}{2} \cdot \epsilon^{-(a'+\ell \cdot b)}$ around them must be disjoint; hence using a simple volume/packing argument we can bound the number of centres from each B_ℓ that fall into the same grid cell by a function of ϵ and d. This helps of establish the first property. We have to do some clean-up to ensure that for each pair $(i, \mathrm{cent}(i)) \in \mathcal{T}$ they belong to the same part (this will imply property 2 of the Theorem) and that at the end for each part P, $|P \cap \mathcal{S}| = |P \cap \mathcal{O}|$. These are a bit technical and are described in details below.

Start by geometrically grouping the centres in $\overline{\mathcal{O}} \cup \overline{\mathcal{S}}$. For $a \in \mathbb{Z}$, let

$$G_a := \left\{ i \in \overline{\mathcal{O}} \cup \overline{\mathcal{S}} : \frac{1}{\epsilon^a} \le D_i < \frac{1}{\epsilon^{a+1}} \right\}.$$

Finally, let $G_{-\infty} := \{i \in \overline{\mathcal{O}} \cup \overline{\mathcal{S}} : D_i = 0\}$. Note that each $i \in \overline{\mathcal{O}} \cup \overline{\mathcal{S}}$ appears in exactly one set among $\{G_a : a \in \mathbb{Z}\} \cup \{G_{-\infty}\}$.

We treat $G_{-\infty}$ differently in our partitioning algorithm. It is important to note that no pair in $\mathcal{T}$ or $\mathcal{N}$ has precisely one point in $G_{-\infty}$, as the following shows.

Lemma 6 *For each pair of centres $(i^*, i) \in \mathcal{T} \cup \mathcal{N}$, $|\{i, i^*\} \cap G_{-\infty}| \ne 1$.*

Proof. Consider some colocated pair $(i^*, i) \in \mathcal{S} \times \mathcal{O}$. As $D_i = D_{i^*} = 0$ and because no other centre in $\mathcal{S} \cup \mathcal{O}$ is colocated with i and i^*, then $i \in \overline{\mathcal{S}}$ and $i^* \in \overline{\mathcal{O}}$. Thus, $\overline{\phi}(i^*) = i$ and it is the unique closest facility in $\overline{\mathcal{O}}$ to i so $i^* = \mathrm{cent}(i)$. This shows every pair $(i^*, i) \in \mathcal{T}$ with either $D_i = 0$ or $D_{i^*} = 0$ must have both $D_i = D_{i^*} = 0$ so $i^*, i \in G_{-\infty}$.

Next consider some $(i^*, i) \in \mathcal{N}$. We know $\epsilon \cdot D_i \le D_{i^*}$ by definition of $\mathcal{N}$. Thus, if $i^* \in G_{-\infty}$ then $i \in G_{-\infty}$ as well. Conversely, suppose $i \in G_{-\infty}$. Since $(i^*, i) \in \mathcal{N}$ then $\delta(i, i^*) \le \epsilon^{-1} \cdot D_i = 0$. So, $D_{i^*} = 0$ meaning $i^* \in G_{-\infty}$ as well. ■

4.1 Partitioning $\{G_a : a \in \mathbb{Z}\}$

Step 1: Forming Bands

Fix b to be the smallest integer that is at least $4/\epsilon$. We first partition $\{G_a : a \in \mathbb{Z}\}$ into *bands*. Sample an integer *shift* a' uniformly at random in the range $\{0, 1, \ldots, b-1\}$. For each $\ell \in \mathbb{Z}$, form the band B_ℓ as follows:

$$B_\ell := \bigcup_{0 \le j \le b-1} G_{a'+j+\ell \cdot b}.$$

Step 2: Cutting Out Cells
Focus on a band B_ℓ. Let $W_\ell := 4d \cdot \epsilon^{-(\ell+2)\cdot b-1}$ (recall that d is the dimension of the Euclidean metric). This will be the "width" of the cells we create. It is worth mentioning that this is the first time we are going to use the properties of Euclidean metrics as all our arguments so far were treating $\delta(\cdot,\cdot)$ as a general metric.

We consider a random grid with cells having width W_ℓ in each dimension. Choose an *offset* $\beta \in \mathbb{R}^d$ uniformly at random from the cube $[0, W_\ell]^d$. For emphasis, we let $\mathbf{p}^i_j$ refer to the j'th component of i's point in Euclidean space (this is the only time we will refer specifically to the coordinates for a point). So centre i has Euclidean coordinates $(\mathbf{p}^i_1, \mathbf{p}^i_2, \ldots, \mathbf{p}^i_d)$.

For any tuple of integers $\mathbf{a} \in \mathbb{Z}^d$ define the *cell* $C^\ell_{\mathbf{a}}$ as follows.

$$C^\ell_{\mathbf{a}} = \{i \in B_\ell : \beta_j + W_\ell \cdot \mathbf{a}_j \leq \mathbf{p}^i_j < \beta_j + W_\ell \cdot (\mathbf{a}_j + 1) \text{ for all } 1 \leq j \leq d\}.$$

These are all points in the band B_ℓ that lie in the half-open cube with side length W_ℓ and lowest corner $\beta + W_\ell \cdot \mathbf{a}$. Each $i \in B_\ell$ lies in precisely one cell as these half-open cubes tile $\mathbb{R}^d$.

Step 3: Fixing $\mathcal{T}$
Let $\mathcal{I} \subseteq \overline{\mathcal{O}}$ be the centres of the form cent(i) for some $(\mathrm{cent}(i), i) \in \mathcal{T}$ where $i, \mathrm{cent}(i) \notin G_{-\infty}$ and i and cent(i) lie in different cells after step 2. We will simply move each $i \in \mathcal{I}$ to the cell containing $\overline{\phi}(i)$. More precisely, for $\ell \in \mathbb{Z}$ and $\mathbf{a} \in \mathbb{Z}$ we define the *part* $P^\ell_{\mathbf{a}}$ as

$$P^\ell_{\mathbf{a}} = (C^\ell_{\mathbf{a}} - \mathcal{I}) \cup \{i \in \mathcal{I} : \overline{\phi}(i) \in C^\ell_{\mathbf{a}}\}.$$

We will show that the parts $P^\ell_{\mathbf{a}}$ essentially satisfy most of the desired properties stated about the random partitioning scheme promised in Theorem 5. It is easy to see that property 2 holds for each part $P^\ell_{\mathbf{a}}$ since for each pair $(\mathrm{cent}(i), i) \in \mathcal{T}$ both of them belong to the same part. All that remains is to ensure they are *balanced* (i.e. have include the same number of centres in $\mathcal{O} \cup \mathcal{S}$) and to incorporate $G_{-\infty}$ and the centres in $\mathcal{O} \cup \mathcal{S}$ that were filtered out. These are relatively easy cleanup steps.

4.2 Properties of the Partitioning Scheme

We will show the parts formed in the partitioning scheme so far have constant size (depending only on ϵ and d) and also that each pair in $\mathcal{N}$ is cut with low probability. Figure 6 illustrates some key concepts in these proofs.

Lemma 7 *For each $\ell \in \mathbb{Z}$ and $\mathbf{a} \in \mathbb{Z}^d$ we have $|P^\ell_{\mathbf{a}}| \leq 2 \cdot (2d)^{2d} \cdot \epsilon^{-9d/\epsilon}$.*

Proof. Note that each centre $i \in C^\ell_{\mathbf{a}}$ witnesses the inclusion of at most one additional centre of $\mathcal{I}$ into $P^\ell_{\mathbf{a}}$ (in particular, cent(i)). So, $|P^\ell_{\mathbf{a}}| \leq 2 \cdot |C^\ell_{\mathbf{a}}|$ meaning it suffices to show $|C^\ell_{\mathbf{a}}| \leq (2d)^{2d} \cdot \epsilon^{-9d/\epsilon}$. For each $i \in C^\ell_{\mathbf{a}}$ we have $D_i \geq \epsilon^{-\ell \cdot b}$. By Lemma 1, the balls of radius $\frac{\epsilon}{2} \cdot \epsilon^{-\ell \cdot b}$ around the centres in $C^\ell_{\mathbf{a}}$ must be disjoint subsets of $\mathbb{R}^d$.

The volume of a ball of radius R in $\mathbb{R}^d$ can be lower bounded by $\left(\frac{\sqrt{2\pi}}{d}\right)^d \cdot R^d$, so the total volume of all of these balls of radius $\frac{\epsilon}{2} \cdot \epsilon^{-\ell \cdot b}$ is at least

$$|C^\ell_{\mathbf{a}}| \cdot \left(\frac{\sqrt{\pi} \cdot \epsilon}{\sqrt{2} \cdot d}\right)^d \cdot \epsilon^{-d \cdot \ell \cdot b}.$$

On the other hand, these balls have their centres located in a common cube with side length W_ℓ and, thus, all such balls are contained in a slightly larger cube with side length $W_\ell + D_i \leq$

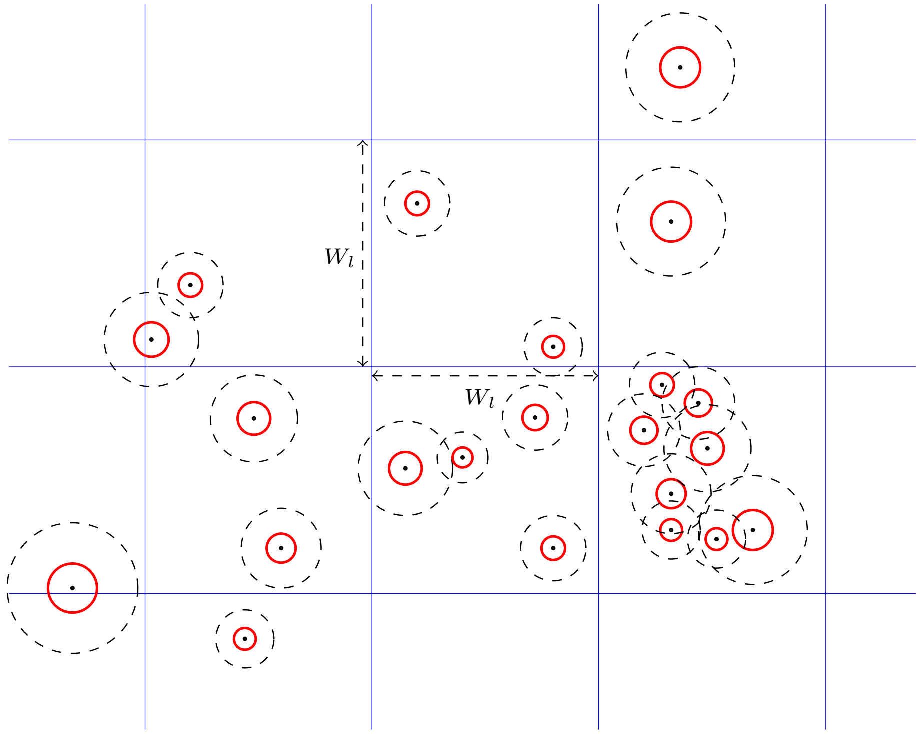

Figure 6: The random grid & balls of radii $\frac{\epsilon \cdot D_i}{2}$ (solid) and $\frac{D_i}{\epsilon}$ (dashed) around each $i \in B_\ell$. Note the balls of radius $\frac{\epsilon \cdot D_i}{2}$ are disjoint, which is used to bound $|C^\ell_{\mathbf{a}}|$ by a volume argument in Lemma 7. For each pair $(i, i^*) \in \mathcal{T} \cup \mathcal{N}$ in the band B_ℓ, one centre must lie in the ball of radius $\frac{D_i}{\epsilon}$ around the other. The proof of Lemma 8 essentially shows that each such ball crosses a grid line with very small probability, so such a pair is probably not cut by the random grid.

$W_\ell + \epsilon^{-(\ell+2)\cdot b+1}$. The total volume of all balls is at most $(W_\ell + \epsilon^{-(\ell+2)\cdot b+1})^d \leq (5d\epsilon^{-(2b+1)})^d \cdot \epsilon^{-d\cdot\ell\cdot b}$. Combining this with the lower bound on the total volume, we see

$$|C^\ell_{\mathbf{a}}| \leq \left(\frac{\sqrt{50} \cdot d^2 \cdot \epsilon^{-(2b+2)}}{\sqrt{\pi}}\right)^d \leq (2d)^{2d} \cdot \epsilon^{-9d/\epsilon}.$$

The last bound follows for sufficiently small ϵ and because b is the smallest integer at least $4/\epsilon$. ∎

Lemma 8 *For each* $(i^*, i) \in \mathcal{N}$ *with* $i, i^* \notin G_{-\infty}$ *we have* $\Pr[i, i^* \text{ lie in different parts}] \leq \epsilon$.

Proof. We first bound the probability they lie in different bands. Note that $D_{i^*} \leq \delta(i, i^*) \leq D_i/\epsilon$ and also $\epsilon D_i \leq D_{i^*}$ because $(i^*, i) \in \mathcal{N}$. Thus, if we say $i \in G_a$ and $i^* \in G_{a^*}$ then $|a - a^*| \leq 1$. The probability that G_a and G_{a^*} are separated when forming the bands B_ℓ is at most $1/b \leq \epsilon/4$.

Next, conditioned on the event that i, i^* lie in the same band B_ℓ, we bound the probability they lie in different cells. Say $i, i^* \in B_\ell$ and note

$$\delta(i, i^*) \leq \epsilon^{-1} \cdot D_i \leq \epsilon^{-(\ell+2)\cdot b} \leq \frac{\epsilon}{4d} \cdot W_\ell.$$

The probability that i and i^* are cut by the random offset along one of the d-dimensions is then at most $\frac{\epsilon}{4d}$. Taking the union bound over all dimensions, the probability that i and i^* lie in different cells is at most $\frac{\epsilon}{4}$.

Finally, we bound the probability that i^* will be moved to a different part when fixing $\mathcal{T}$. If $(i^*, \overline{\phi}(i^*)) \notin \mathcal{T}$ or if $i^* \notin \mathcal{I}$ then this will not happen. So, we will bound $\Pr[i^* \in \mathcal{I}]$ if $(i^*, \overline{\phi}(i^*)) \in \mathcal{T}$. That is, we bound the probability that i', i^* lie in different parts where i' is such that $i^* = \mathrm{cent}(i')$ and $(i^*, i') \in \mathcal{T}$. Note that $D_{i'} \leq \delta(i', i^*) \leq (1+\epsilon) \cdot D_{i^*} \leq \epsilon^{-1} \cdot D_{i^*}$ by Lemma 2 and $D_{i'} \geq \epsilon \cdot \delta(i', i^*) = \epsilon \cdot D_{i^*}$ by definition of $\mathcal{T}$. So i' and i^* lie in different bands with probability at most $\frac{\epsilon}{4}$ by the same argument as with (i, i^*). Similarly, conditioned on i', i^* lying in the same band, the probability they lie in different cells is at most $\frac{\epsilon}{4}$ again by the same arguments as with (i, i^*).

Note that if i, i^* lie in different parts then they were cut by the random band or by the random box, or else i', i^* were cut by the random band or the random box (the latter may not apply if i^* is not involved in a pair in $\mathcal{T}$). Thus, by the union bound we have

$$\begin{aligned}
\Pr[i, i^* \text{ are in different parts}] \quad \leq \quad & \Pr[i, i^* \text{ lie in different } B_\ell] + \\
& \Pr[i, i^* \text{ lie in different } C^\ell_{\mathbf{a}} \mid i, i^* \text{ lie in the same } B_\ell] + \\
& \Pr[i^*, i' \text{ lie in different } B_\ell] + \\
& \Pr[i^*, i' \text{ lie in different } C^\ell_{\mathbf{a}} \mid i^*, i' \text{ lie in the same } B_\ell] \\
\leq \quad & \frac{\epsilon}{4} + \frac{\epsilon}{4} + \frac{\epsilon}{4} + \frac{\epsilon}{4} = \epsilon
\end{aligned}$$

■

4.3 Balancing the Parts

We have partitioned $\{G_a : a \in \mathbb{Z}\}$ into parts $P^\ell_{\mathbf{a}}$ for various $\ell \in \mathbb{Z}$ and $\mathbf{a} \in \mathbb{Z}^d$. We extend this to a partition of all of $\mathcal{O} \cup \mathcal{S}$ into constant-size parts that are balanced between $\mathcal{O}$ and $\mathcal{S}$ to complete the proof of Theorem 5. Let $\mathcal{P} = \{P^\ell_{\mathbf{a}} : \ell \in \mathbb{Z}, \mathbf{a} \in \mathbb{Z}^d \text{ and } P^\ell_{\mathbf{a}} \neq \emptyset\}$ be the collection of nonempty parts formed so far.

The proof of Lemma 6 shows that $G_{-\infty}$ partitions naturally into colocated centres. So let $\mathcal{P}_{-\infty}$ denote the partition of $G_{-\infty}$ into these pairs. Finally let $\mathcal{P}'$ denote the partition of $(\mathcal{O} - \overline{\mathcal{O}}) \cup (\mathcal{S} - \overline{\mathcal{S}})$ into singleton sets.

We summarize important properties of $\mathcal{P} \cup \mathcal{P}_{-\infty} \cup \mathcal{P}'$.

- $\mathcal{P} \cup \mathcal{P}_{-\infty} \cup \mathcal{P}'$ is itself a partitioning of $\mathcal{O} \cup \mathcal{S}$ into parts with size at most $2 \cdot (2d)^{2d} \cdot \epsilon^{-9d/\epsilon}$ (Lemma 7).
- Every $(\mathrm{cent}(i), i) \in \mathcal{T}$ pair has both endpoints in the same part. This is true for pairs with endpoints in $\mathcal{P}$ by Step 3 in the partitioning of $\{G_a : a \in \mathbb{Z}\}$. If $i \in G_{-\infty}$ this is trivially true as colocated pairs $(i^*, i) \in \overline{\mathcal{O}} \cup \overline{\mathcal{S}}$ must have $i^* = \mathrm{cent}(i)$.
- Over the randomized formation of $\mathcal{P}$, each $(i^*, i) \in \mathcal{N}$ has endpoints in different parts with probability at most ϵ. For i, i^* lying in $\mathcal{P}$ this follows from Lemma 8. For i, i^* lying in $\mathcal{P}_{-\infty}$ this follows because they must then be colocated pairs so they always form a part by themselves.

From now on, we simply let $\overline{\mathcal{P}} = \mathcal{P} \cup \mathcal{P}_{-\infty} \cup \mathcal{P}'$. We show how to combine parts of $\overline{\mathcal{P}}$ into constant-size parts that are also balanced between $\mathcal{O}$ and $\mathcal{S}$. Since merging parts does not destroy the property of two centres lying together, we will still have that pairs of centres in $\mathcal{T}$ appear together and any pair of centres in $\mathcal{N}$ lie in different parts with probability at most ϵ.

For any subset $A \subseteq \mathcal{O} \cup \mathcal{S}$, let $\mu(A) = |A \cap \mathcal{O}| - |A \cap \mathcal{S}|$ denote the *imbalance* of A.

Lemma 9 *Let $Y \geq 1$ be an integer and $\mathcal{A}$ a collection of disjoint, nonempty subsets of $\mathcal{O} \cup \mathcal{S}$ such that $\sum_{A \in \mathcal{A}} \mu(A) = 0$. If $|A| \leq Y$ for each A then there is some nonempty $\mathcal{B} \subseteq \mathcal{A}$ where $|\mathcal{B}| \leq 2Y^3$ such that $\sum_{B \in \mathcal{B}} \mu(B) = 0$.*

Proof. If $\mu(A) = 0$ for some $A \in \mathcal{A}$ then simply let $\mathcal{B} = \{A\}$. Otherwise, note $|\mu(A)| \leq |A| \leq Y$ for each $A \in \mathcal{A}$ and partition $\mathcal{A}$ into sets $\mathcal{A}_x := \{A \in \mathcal{A} : \mu(A) = x\}$ for $x \in \{1, \ldots, Y\} \cup \{-1, \ldots, -Y\}$. We have

$$\sum_{x=1}^{Y} |\mathcal{A}_{-x}| \leq \sum_{x=1}^{Y} |\mathcal{A}_{-x}| \cdot x = \sum_{x=1}^{Y} |\mathcal{A}_{x}| \cdot x \leq Y \cdot \sum_{x=1}^{Y} |\mathcal{A}_{x}|,$$

so if $\sum_{x=1}^{Y} |A_x| \leq Y^2$ then $|\mathcal{A}| \leq 2Y^3$ and we can take $\mathcal{B} = \mathcal{A}$. Similarly if $\sum_{x=1}^{Y} |\mathcal{A}_{-x}| \leq Y^2$ then we can take $\mathcal{B} = \mathcal{A}$.

Finally, we are left with the case that $\sum_{x=1}^{Y} |\mathcal{A}_x|$ and $\sum_{x=1}^{Y} |\mathcal{A}_{-x}|$ both exceed Y^2. By the pigeonhole principle, there are values $1 \leq x, y \leq Y$ such that $|\mathcal{A}_x| \geq Y$ and $|\mathcal{A}_{-y}| \geq Y$. In this case, we take $\mathcal{B}$ to be any y sets from $\mathcal{A}_x$ plus any x sets from $\mathcal{A}_{-y}$. Then $|\mathcal{B}| = x + y \leq 2Y$ and $\mathcal{B}$ is balanced, which is what we needed to show. ■

To complete the partitioning we iteratively apply Lemma 9 to $\overline{\mathcal{P}}$ with $Y = 2 \cdot (2d)^{2d} \cdot \epsilon^{-9d/\epsilon}$ where we note $|P| \leq Y$ by Lemma 7. Each iteration, we find some nonempty $\mathcal{Q} \subseteq \overline{\mathcal{P}}$ such that $\sum_{Q \in \mathcal{Q}} \mu(Q) = 0$. Remove $\mathcal{Q}$ from $\overline{\mathcal{P}}$ and repeat until all sets from $\overline{\mathcal{P}}$ have been removed. Each balanced part obtained is the union of at most $2 \cdot \left(2 \cdot (2d)^{2d} \cdot \epsilon^{-9d/\epsilon}\right)^3$ different parts in $\overline{\mathcal{P}}$, meaning each part has size at most $2 \cdot \left(2 \cdot (2d)^{2d} \cdot \epsilon^{-9d/\epsilon}\right)^4 = 32 \cdot (2d)^{8d} \cdot \epsilon^{-36 \cdot d/\epsilon}$.

5 Proof of Theorem 1 and Extension to ℓ_q^q-norm k-clustering

In this section we show how to extended the analysis presented in the previous sections for $\mathbb{R}^d$ to prove Theorem 1. For doubling metrics we use $\rho = \rho(\epsilon, d) = 32 \cdot d^{16d} \cdot \epsilon^{-256d/\epsilon}$ and consider the ρ-swap local search. First we argue why we get a PTAS when the metric $\delta(\cdot, \cdot)$ is a doubling metric.

5.1 Extending to Doubling Metrics

Let (V, δ) be a doubling metric space with doubling dimension d (so each ball of radius $2r$ can be covered with 2^d balls of radius r in V). Note that the only places in the analysis where we used the properties of metric being $\mathbb{R}^d$ was in the proof of Theorem 5 and in particular in Step 2 where we cut out cells from bands and then later in the proof of Lemma 7. The rest of the proof remains unchanged.

Given a metric (V, δ), the aspect ratio, denoted by Δ, is the ratio of the largest distance to the smallest non-zero distance in the metric: $\Delta = \frac{\max_{u,v \in V} \delta(u,v)}{\min_{u,v \in V, u \neq v} \delta(u,v)}$. Talwar [49] gave a hierarchical decomposition of a doubling metric using an algorithm similar to one by Fakcharoenphol et al. [24]. It assumes $\min_{u,v \in V, u \neq v} \delta(u, v) = 1$, which can be accomplished by scaling the distances. So, Δ is the maximum distance between two points in the metric.

Theorem 6 *[49] Suppose $\min_{u,v \in V : u \neq v} \delta(u, v) = 1$. There is a randomized hierarchical decomposition of V, which is a sequence of partitions $\mathcal{P}_0$, $\mathcal{P}_1$, $\ldots$, $\mathcal{P}_h$, where $\mathcal{P}_{i-1}$ is a refinement of $\mathcal{P}_i$, $\mathcal{P}_h = \{V\}$, and $\mathcal{P}_0 = \{\{v\}\}_{v \in V}$. The decomposition has the following properties:*

1. *$\mathcal{P}_0$ corresponds to the leaves and $\mathcal{P}_h$ corresponds to the root of the split-tree T, and the height of T is $h = \varphi + 2$, where $\varphi = \log \Delta$ and Δ is the aspect ratio of metric.*

2. *For each level i and each $S \in \mathcal{P}_i$, S has diameter at most 2^{i+1}.*

3. *The branching factor b of T is at most 12^d.*

4. For any $u, v \in V$, the probability that they are in different sets corresponding to nodes in level i of T is at most $5d \cdot 2^{-i} \cdot \delta(u,v)$.

The proof of randomized partitioning scheme for the case of doubling metrics follows similarly as in Section 4. We create bands B_ℓ as before. For each band B_ℓ we cut out the cells (that define the parts) in the following way. First, observe that for all $i \in B_\ell$ we have $\frac{1}{\epsilon^{\ell b}} \leq D_i < \frac{1}{\epsilon^{(\ell+2)b-1}}$. Additionally, by Lemma 1 for all $i \in \overline{\mathcal{S}} \cap B_\ell$, $i^* \in \overline{\mathcal{O}} \cap B_\ell$ we have $\frac{1}{\epsilon^{\ell b-1}} \leq \delta(i,i^*)$. So, if we scale all distances between points in B_ℓ so the minimum distance is 1, for all $i \in B_\ell$ we have $1 \leq \epsilon \cdot D_i < \frac{1}{\epsilon^{2b}}$ and for any two distinct $i, i^* \in B_\ell$ we have $1 \leq \delta(i,i^*)$.

Consider the hierarchical decomposition of the points in B_ℓ using Theorem 6 in this scaled metric and consider the clusters defined by the sets at level $\lambda = \log d + (2b+3)\log\frac{1}{\epsilon}$. These will define our cells for B_ℓ, let's call them $C^\ell_1, \ldots, C^\ell_q$.[1] The only two things that we need to do is: a) bound the size of each cell C^ℓ_r and b) to show that Lemma 8 still holds. Bounding $|C^\ell_r|$ will be equivalent to Lemma 7. We then we apply Step 3 to fix $\mathcal{T}$ as before, this will at most double the size of each cell. The rest of the proof of Theorem 5 remains unchanged as it does not rely on the geometry of $\mathbb{R}^d$.

To bound $|C^\ell_i|$ we use property 3 of Theorem 6 and the fact that the bottom level $\mathcal{P}_0$ consists of singleton sets:

$$\begin{aligned} |C^\ell_i| &\leq 12^{d\cdot\lambda} \\ &= 2^{(\log 12)d\cdot\left[(2b+3)\log\frac{1}{\epsilon}+\log d\right]} \\ &\leq \epsilon^{-64d/\epsilon}\cdot d^{4d}, \end{aligned}$$

where we used the definition of b in the last inequality. Using similar argument as in the case of $\mathbb{R}^d$, we can bound the size of each part by $4Y^4$ where $Y \leq 2|C^\ell_i|$, which implies an upper bound of $32 \cdot d^{16d} \cdot \epsilon^{-256d/\epsilon}$.

As for proving equivalent version of Lemma 8, it is enough to show that $\Pr[i, i^*$ lie in different $C^\ell_r]$ given that they are in the same band B_ℓ is at most $\epsilon/4$ since the rest of that proof is the same. For such pair i, i^*, note that $\delta(i,i^*) \leq \epsilon^{-1}\cdot D_i \leq \epsilon^{-2b-1}$. Using property 4 of Theorem 6:

$$\begin{aligned} \Pr[i, i^* \text{ lie in different } C^\ell_{\mathbf{r}} \mid i, i^* \text{ lie in the same } B_\ell] &\leq 5d\frac{\delta(i,i^*)}{2^\lambda} \\ &\leq 5d\frac{\epsilon^{-2b-1}}{d\epsilon^{-2b-3}} \\ &= 5\epsilon^2 \\ &\leq \epsilon/4. \end{aligned}$$

The rest of the proof of Theorem 5 remains the same as in Subsection 4.3.

5.2 Extending to ℓ_q^q-norm k-clustering

It is fairly straightfoward to see that all the calculations in the proof of Theorem 4 where we bound Δ^P_j will hold with similar bounds if the objective function is sum of distances (instead of squares of

[1] It is easy to see that we really don't need the full hierarchical decomposition as in Theorem 6. One can simply run one round of the *Decompose* algorithm of [49] that carves out clusters with diameter size at most 2^λ and show that the relevant properties of Theorem 6 hold. For ease of presentation we simply invoke Theorem 6.

Algorithm 3 ρ-Swap Local Search for UNCAPACITATED FACILITY LOCATION

Let $\mathcal{S} \leftarrow \mathcal{C}$
while $\exists$ set $\emptyset \subsetneq \mathcal{S}' \subseteq \mathcal{C}$, $|\mathcal{S} - \mathcal{S}'|, |\mathcal{S}' - \mathcal{S}| \leq \rho$ s.t. $\text{cost}_{\mathbf{ufl}}(\mathcal{S}') < \text{cost}_{\mathbf{ufl}}(\mathcal{S})$ **do**
  $\mathcal{S} \leftarrow \mathcal{S}'$
return $\mathcal{S}$

distances), that is ℓ_1-clustering (i.e. k-MEDIAN). For the case of ℓ_q^q (with $q > 2$) it is also possible to verify that whenever we have a term of the form $O(\epsilon)(c_j^* + c_j)$ when upper bounding Δ_j^P in the analysis of k-MEANS (e.g. when j is lucky, or long, or good) then the equivalent term for ℓ_q^q will be $O(\epsilon 2^q)(\delta_j^q + \delta_j^{*q})$. For the case when j is bad we still get an upper bound of $\Delta_j^P \leq O(2^q(\delta_j^q + \delta_j^{*q}))$. One can use the crude bound of $(x + \epsilon \cdot y)^q \leq x^q + 2^q\epsilon \cdot \max\{x, y\}$ for all $x, y \geq 0$ to help bound terms like $(d_j + \epsilon \cdot D_{\sigma(j)})^q$. We skip the straightforward (but tedious) details.

This means we get a $(1 + O(\epsilon))$-approximation for k-clustering when the objective function is measured according to the ℓ_q^q-norm for any fixed $q \geq 1$ by considering swaps of size $\rho = d^{O(d)} \cdot (2^q/\epsilon)^{-d \cdot 2^q/\epsilon}$ in the local search algorithm.

6 Extending to the uncapacitated facility location and generalized k-median problems

Recall that the setting of UNCAPACITATED FACILITY LOCATION is similar to k-MEDIAN except we are given opening costs for each $i \in \mathcal{C}$ instead of a cardinality bound k. A feasible solution is any nonempty $\mathcal{S} \subseteq \mathcal{C}$ and its cost is

$$\text{cost}_{\mathbf{ufl}}(\mathcal{S}) = \sum_{j \in \mathcal{X}} \delta(j, \mathcal{S}) + \sum_{i \in \mathcal{S}} f_i.$$

The standard local search algorithm is the following. Let ρ be defined as before.

We will show every locally optimum solution has cost at most $(1 + O(\epsilon)) \cdot OPT$ using at most $2 \cdot |\mathcal{C}|$ test swaps. Thus, replacing the cost condition in Algorithm 1 with $\text{cost}_{\mathbf{ufl}}(\mathcal{S}') \leq \left(1 - \frac{\epsilon}{2 \cdot |\mathcal{C}|}\right) \cdot \text{cost}_{\mathbf{ufl}}(\mathcal{S})$ is a polynomial-time variant that is also a PTAS.

Let $\mathcal{S}$ be a locally optimum solution and $\mathcal{O}$ a global optimum. We may assume, by duplicating points if necessary (for the analysis only), that $\mathcal{S} \cap \mathcal{O} = \emptyset$, thus $|\mathcal{S}| + |\mathcal{O}| \leq 2 \cdot |\mathcal{C}|$ (where $|\mathcal{C}|$ refers to the size of the original set of facilities before duplication). Our analysis proceeds in a manner that is nearly identical to our approach for k-MEANS.

In particular, we prove the following. It is identical to Theorem 5 in every way except the first point does not require $P \cap \mathcal{O}$ and $P \cap \mathcal{S}$ to have equal size.

Theorem 7 *There is a randomized algorithm that samples a partitioning π of $\mathcal{O} \cup \mathcal{S}$ such that:*

- *For each part $P \in \pi$, $|P \cap \mathcal{O}|, |P \cap \mathcal{S}| \leq \rho(\epsilon, d)$.*
- *For each part $P \in \pi$, $\mathcal{S} \triangle P$ includes at least one centre from every pair in $\mathcal{T}$.*
- *For each $(i^*, i) \in \mathcal{N}$, $\Pr[i, i^*$ lie in different parts of $\pi] \leq \epsilon$.*

The proof is identical to Theorem 5, except we do not to the "balancing" step in Section 4.3. Indeed this was the only step in the proof of Theorem 5 that required $|\mathcal{S}| = |\mathcal{O}|$.

We now complete the proof of Theorem 2.

Proof. Let $c_j = \delta(j, \mathcal{S})$ and $c_j^* = \delta(j, \mathcal{O})$ for each $j \in \mathcal{X}$. Sample a partition π of $\mathcal{S} \cup \mathcal{O}$ as per

Algorithm 4 ρ-Swap Local Search for UNCAPACITATED FACILITY LOCATION

Let $\mathcal{S}$ be any k centres from $\mathcal{S}$
while $\exists$ set $\mathcal{S}' \subseteq \mathcal{C}$, with $|\mathcal{S} - \mathcal{S}'|, |\mathcal{S}' - \mathcal{S}| \leq \rho$ and $1 \leq |\mathcal{S}'| \leq k$ s.t. $\text{cost}_{\mathbf{gkm}}(\mathcal{S}') < \text{cost}_{\mathbf{gkm}}(\mathcal{S})$ **do**
  $\mathcal{S} \leftarrow \mathcal{S}'$
return $\mathcal{S}$

Theorem 7. For each part $P \in \pi$ and each $j \in \mathcal{X}$ we let Δ_j^P denote $\delta(j, \mathcal{S} \triangle P) - \delta(j, \mathcal{S})$. Using the same bounds considered in the proof of Theorem 4 we have

$$\mathbf{E}_\pi \left[\sum_{P \in \pi} \Delta_j^p \right] \leq (1 + O(\epsilon)) \cdot c_j^* - (1 - O(\epsilon)) c_j.$$

As each $i \in \mathcal{S}$ is closed exactly once and each $i^* \in \mathcal{O}$ is opened exactly once, the fact that $\mathcal{S}$ is locally optimal and fact that each $P \in \pi$ has $|P \cap \mathcal{S}|, |P \cap \mathcal{O}| \leq \rho$ shows

$$\begin{aligned} 0 &\leq \mathbf{E}_\pi \left[\sum_{P \in \pi} \text{cost}_{\mathbf{ufl}}(\mathcal{S} \triangle P) - \text{cost}_{\mathbf{ufl}}(\mathcal{S}) \right] \\ &\leq \left(\sum_{j \in \mathcal{X}} (1 + O(\epsilon)) \cdot c_j^* - (1 - O(\epsilon)) c_j \right) + \left(\sum_{i^* \in \mathcal{O}} f_{i^*} - \sum_{i \in \mathcal{S}} f_i \right). \end{aligned}$$

Rearranging shows $\text{cost}_{\mathbf{ufl}}(\mathcal{S}) \leq (1 + O(\epsilon)) \text{cost}_{\mathbf{ufl}}(\mathcal{O})$. ■

Finally we conclude by analyzing a local search algorithm for GENERALIZED k-MEDIAN. Here we are given both a cardinality bound k and opening costs for each $i \in \mathcal{C}$. The goal is to open some $\mathcal{S} \subseteq \mathcal{C}$ with $1 \leq |\mathcal{S}| \leq k$ to minimize

$$\text{cost}_{\mathbf{gkm}}(\mathcal{S}) = \sum_{j \in \mathcal{X}} \delta(j, \mathcal{S}) + \sum_{i \in \mathcal{S}} f_i.$$

Note this is the same as $\text{cost}_{\mathbf{ufl}}$, we use this slightly different notation to avoid confusion as we are discussing a different problem now.

We can prove Theorem 3 very easily.

Proof. Let $\mathcal{S}$ and $\mathcal{O}$ denote a local optimum and a global optimum (respectively). By adding artificial points i with $f_i = 0$ that are extremely far away from $\mathcal{X}$, we may assume $\mathcal{S} = \mathcal{O}$. Then we simply use our original partitioning result, Theorem 5, to define test swaps. Noting that each $i \in \mathcal{S}$ is closed exactly once and each $i^* \in \mathcal{O}$ is opened exactly once over the swaps induced by a partition π, we proceed in the same way as above in the proof of Theorem 2 and see

$$\text{cost}_{\mathbf{gkm}}(\mathcal{S}) \leq (1 + O(\epsilon)) \cdot \text{cost}_{\mathbf{gkm}}(\mathcal{O}).$$

■

Using almost identical arguments as we did in Section 5.2 for extension of k-MEANS to ℓ_q^q-norm in doubling dimensions, one can easily extend the results of Theorem 2 and 3 to the case where we consider the sum of q'th power of distances (ℓ_q^q-norm).

7 Conclusion

We have presented a PTAS for k-MEANS in constant-dimensional Euclidean metrics and, more generally, in doubling metrics and when the objective is to minimizing the ℓ_q^q-norm of distances between points and their nearest centres for any constant $q \geq 1$. This is also the first approximation for k-MEDIAN in doubling metrics and the first demonstration that local search yields a true PTAS for k-MEDIAN even in the Euclidean plane. Our approach extends to showing local search yields a PTAS for UNCAPACITATED FACILITY LOCATION and GENERALIZED k-MEDIAN in doubling metrics.

The running time of a single step of the local search algorithm is $O(k^\rho)$ where $\rho = d^{O(d)} \cdot \epsilon^{-O(d/\epsilon)}$ for the case of k-MEANS when the metric has doubling dimension d. We have not tried to optimize the constants in the $O(\cdot)$ notations in ρ. The dependence on d cannot be improved much under the Exponential Time Hypothesis (**ETH**). For example, if the running time of a single iteration was only $O(k^{\exp(d^{1-\delta}) \cdot f(\epsilon)})$ for some constant δ then we would have a sub-exponential time $(1+\epsilon)$-approximation when $d = O(\log n)$. Recall that k-MEANS is **APX**-hard when $d = \Theta(\log n)$ [10], so this would refute the **ETH**.

It may still be possible to obtain an EPTAS for any constant dimension d. That is, there could be a PTAS with running time of the form $O(g(\epsilon) \cdot n^{\exp(d)})$ for some function $g(\epsilon)$ (perhaps depending also on d). Finally, what is the fastest PTAS we can obtain in the special case of the Euclidean plane (i.e. $d = 2$)? It would be interesting to see if there is an EPTAS whose running time is linear or near linear in n for any fixed constant ϵ.

References

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

[2] Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998.

[3] Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. Approximation schemes for Euclidean K-medians and related problems. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing (STOC '98)*, pages 106–113. ACM, 1998.

[4] David Arthur, Bodo Manthey, and Heiko Röglin. Smoothed analysis of the K-means method. *J. ACM*, 58(5):19:1–19:31, 2011.

[5] David Arthur and Sergei Vassilvitskii. How slow is the K-means method? In *Proceedings of the Twenty-second Annual Symposium on Computational Geometry (SoCG '06)*, pages 144–153. ACM, 2006.

[6] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '07)*, pages 1027–1035. SIAM, 2007.

[7] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristic for K-median and facility location problems. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing (STOC '01)*, pages 21–29. ACM, 2001.

[8] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for K-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.

[9] Pranjal Awasthi, Avrim Blum, and Or Sheffet. Stability yields a PTAS for K-median and K-means clustering. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS '10)*, pages 309–318. IEEE Computer Society, 2010.

[10] Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The Hardness of Approximation of Euclidean K-Means. In *Proceedings of 31st International Symposium on Computational Geometry (SoCG '15)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 754–767, 2015.

[11] Mihai Bādoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via Coresets. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing (STOC '02)*, pages 250–257. ACM, 2002.

[12] Sayan Bandyapadhyay and Kasturi Varadarajan. On variants of K-means clustering. In *Proceedings of the 32nd International Symposium on Computational Geometry (SoCG '16)*, Leibniz International Proceedings in Informatics (LIPIcs), 2016.

[13] Johannes Blömer, Christiane Lammersen, Melanie Schmidt, and Christian Sohler. Theoretical analysis of the K-means algorithm - A survey. *CoRR*, abs/1602.08254, 2016.

[14] Jaroslaw Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for K-median, and positive correlation in budgeted optimization. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '15)*, pages 737–756. SIAM, 2015.

[15] Ke Chen. On Coresets for K-median and K-means clustering in metric and Euclidean spaces and their applications. *SIAM J. Comput.*, 39:923–947, 2009.

[16] Vincent Cohen-Addad, Philip N. Klein, and Claire Mathieu. Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 353–364, 2016.

[17] Vincent Cohen-Addad, Philip N. Klein, and Claire Mathieu. The power of local search for clustering. *CoRR*, abs/1603.09535, 2016.

[18] Vincent Cohen-Addad and Claire Mathieu. Effectiveness of local search for geometric optimization. In *Proceedings of the 31st International Symposium on Computational Geometry (SoCG '15)*, Leibniz International Proceedings in Informatics (LIPIcs), 2015.

[19] Vincent Cohen-Addad and Claire Mathieu. Effectiveness of local search for geometric optimization. In *Proceedings of 31st Annual Symposium on Computational Geometry (SOCG '15)*, pages 329–344, 2015.

[20] Sanjoy Dasgupta. How fast is K-means? In *Proceedings of the 16th Annual Conference on Learning Theory and 7th Kernel Workshop (COLT/Kernel '03)*, pages 735–735, 2003.

[21] W. Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani. Approximation schemes for clustering problems. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing (STOC '03)*, pages 50–58. ACM, 2003.

[22] Patrik D'haeseleer et al. How does gene expression clustering work? *Nature biotechnology*, 23(12):1499–1502, 2005.

[23] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Mach. Learn.*, 56(1-3):9–33, 2004.

[24] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing (STOC '03)*, pages 448–455. ACM, 2003.

[25] Dan Feldman, Morteza Monemizadeh, and Christian Sohler. A PTAS for K-means clustering based on weak Coresets. In *Proceedings of the Twenty-third Annual Symposium on Computational Geometry (SoCG '07)*, SoCG '07, pages 11–18. ACM, 2007.

[26] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size Coresets for K-means, PCA and projective clustering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '13)*, pages 1434–1453. SIAM, 2013.

[27] Zachary Friggstad, Mohsen Rezapour, and Mohammad R. Salavatipour. Local search yields a ptas for k-means in doubling metrics. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 365–374, 2016.

[28] Zachary Friggstad, Mohsen Rezapour, and Mohammad R. Salavatipour. Local search yields a PTAS for k-means in doubling metrics. *CoRR*, abs/1603.08976, 2016.

[29] A. Gupta and T. Tangwongsan. Simpler analyses of local search algorithms for facility location. *CoRR, abs/0809.2554*, 2008.

[30] Sariel Har-Peled and Akash Kushal. Smaller Coresets for K-median and K-means clustering. In *Proceedings of the Twenty-first Annual Symposium on Computational Geometry (SoCG '05)*, pages 126–134. ACM, 2005.

[31] Sariel Har-Peled and Soham Mazumdar. On Coresets for K-means and K-median clustering. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing (STOC '04)*, pages 291–300. ACM, 2004.

[32] Sariel Har-Peled and Bardia Sadri. How fast is the K-means method? *Algorithmica*, 41(3):185–202, 2005.

[33] Matan Hofree, John P Shen, Hannah Carter, Andrew Gross, and Trey Ideker. Network-based stratification of tumor mutations. *Nature methods*, 10(11):1108–1115, 2013.

[34] Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted Voronoi diagrams and randomization to variance-based K-clustering. In *Proceedings of the tenth annual Symposium on Computational Geometry (SoCG '94)*, pages 332–339. ACM, 1994.

[35] Anil K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recogn. Lett.*, 31(8):651–666, 2010.

[36] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for K-means clustering. *Comput. Geom. Theory Appl.*, 28(2-3):89–112, 2004.

[37] Stavros G. Kolliopoulos and Satish Rao. A nearly linear-time approximation scheme for the Euclidean Kappa-median problem. In *Proceedings of the 7th Annual European Symposium on Algorithms (ESA '99)*, pages 378–389. Springer-Verlag, 1999.

[38] Amit Kumar and Ravindran Kannan. Clustering with spectral norm and the K-means algorithm. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS '10)*, pages 299–308. IEEE Computer Society, 2010.

[39] Amit Kumar, Yogish Sabharwal, and Sandeep Sen. A simple linear time $(1+\epsilon)$ -approximation algorithm for K-means clustering in any dimensions. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS '04)*, pages 454–462. IEEE Computer Society, 2004.

[40] Amit Kumar, Yogish Sabharwal, and Sandeep Sen. Linear-time approximation schemes for clustering problems in any dimensions. *J. ACM*, 57(2):5:1–5:32, 2010.

[41] Michael Langberg and Leonard J. Schulman. Universal ϵ-approximators for integrals. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '10)*, pages 598–607. SIAM, 2010.

[42] Shi Li. A 1.488-approximation for the uncapacitated facility location problem. In *Proceedings of the 38th Annual International Colloquium on Automata, Languages and Programming (ICALP '11)*, pages 45–58, 2011.

[43] Shi Li and Ola Svensson. Approximating K-median via pseudo-approximation. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing (STOC '13)*, pages 901–910. ACM, 2013.

[44] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*, 28(2):129–137, 2006.

[45] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar K-means problem is NP-hard. In *Proceedings of the 3rd International Workshop on Algorithms and Computation (WALCOM '09)*, pages 274–285. Springer-Verlag, 2009.

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

[47] Rafail Ostrovsky and Yuval Rabani. Polynomial-Time Approximation Schemes for geometric min-sum median clustering. *J. ACM*, 49(2):139–156, 2002.

[48] Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Chaitanya Swamy. The effectiveness of Lloyd-type methods for the K-means problem. *J. ACM*, 59(6):28:1–28:22, 2013.

[49] Kunal Talwar. Bypassing the embedding: Algorithms for low dimensional metrics. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing (STOC '04)*, pages 281–290. ACM, 2004.

[50] Andrea Vattani. The hardness of K-means clustering in the plane. *Manuscript*, 2009.

[51] Andrea Vattani. K-means requires exponentially many iterations even in the plane. *Discrete Comput. Geom.*, 45(4):596–616, 2011.