



Faster Approximation Schemes for (Constrained) k -Means with Outliers

Zhen Zhang  

School of Advanced Interdisciplinary Studies, Hunan University of Technology and Business,
Changsha, China
Xiangjiang Laboratory, Changsha, China

Junyu Huang  

School of Computer Science and Engineering, Central South University, Changsha, China

Qilong Feng  

School of Computer Science and Engineering, Central South University, Changsha, China

Abstract

Given a set of n points in \mathbb{R}^d and two positive integers k and m , the Euclidean k -means with outliers problem aims to remove at most m points, referred to as outliers, and minimize the k -means cost function for the remaining points. Developing algorithms for this problem remains an active area of research due to its prevalence in applications involving noisy data. In this paper, we give a $(1 + \varepsilon)$ -approximation algorithm that runs in $n^2 d ((k + m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ time for the problem. When combined with a coresets construction method, the running time of the algorithm can be improved to be linear in n . For the case where k is a constant, this represents the first polynomial-time approximation scheme for the problem: Existing algorithms with the same approximation guarantee run in polynomial time only when both k and m are constants. Furthermore, our approach generalizes to variants of k -means with outliers incorporating additional constraints on instances, such as those related to capacities and fairness.

2012 ACM Subject Classification Theory of computation \rightarrow Facility location and clustering

Keywords and phrases Approximation algorithms, clustering

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.84

Funding This work was supported by National Natural Science Foundation of China (62202161, 62172446), Natural Science Foundation of Hunan Province (2023JJ40240), and Scientific Research Fund of Hunan Provincial Education Department (23B0597).

1 Introduction

Clustering is a frequently encountered task in many fields related to machine learning, aiming to partition a given set of points into several cohesive clusters. Among the various ways of formalizing the task of clustering, the Euclidean k -means problem is perhaps the most commonly studied one. In this problem, we are given a set $\mathcal{P} \subset \mathbb{R}^d$ of points and a positive integer k , and the goal is to identify a set $\mathcal{C} \subset \mathbb{R}^d$ of no more than k centers so that the objective function $\sum_{p \in \mathcal{P}} \min_{c \in \mathcal{C}} \|p - c\|^2$ is minimized. Here, the points are partitioned into different clusters according to the disparities in their corresponding centers, namely, the nearest ones. In most applications of the problem, the upper bound on the number of centers (i.e., k) is significantly smaller than the number of points to be clustered. This prompts considerable efforts in developing algorithms for the case where k is fixed. Specifically, it is known that the problem admits *polynomial-time approximation schemes* (PTASs) when k is a constant [14, 18, 11, 37, 33, 34].

Despite extensive study, algorithms developed for the Euclidean k -means problem often exhibit poor performance. The main issue lies in the lack of robustness of the objective function to noisy data: A few *outliers* within the point set can significantly impact the



© Zhen Zhang, Junyu Huang, and Qilong Feng;
licensed under Creative Commons License CC-BY 4.0

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).

Editors: Rastislav Kráľovič and Antonín Kučera; Article No. 84; pp. 84:1–84:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

value of the function. Removing these outliers typically leads to a better clustering result. Motivated thus, we consider the Euclidean k -means with outliers (k -MEANSOUT) problem, which can be defined as follows.

► **Definition 1** (Euclidean k -MEANSOUT). *An instance of Euclidean k -MEANSOUT is specified by a set $\mathcal{P} \subset \mathbb{R}^d$ of points and two positive integers k and m . A feasible solution to the instance is a set $\mathcal{C} \subset \mathbb{R}^d$ of centers satisfying $|\mathcal{C}| \leq k$ and a set $\mathcal{O} \subseteq \mathcal{P}$ of outliers satisfying $|\mathcal{O}| \leq m$. The cost of such a solution is $\sum_{p \in \mathcal{P} \setminus \mathcal{O}} \min_{c \in \mathcal{C}} \|p - c\|^2$. The goal of Euclidean k -MEANSOUT is to find a feasible solution with minimum cost.*

Solving the Euclidean k -MEANSOUT problem helps to the removal of more interpretable outliers that can be contextualized by the clusters. This, in turn, results in more cohesive clusters. Notably, it has been observed that adopting this joint perspective on outlier detection and clustering leads to improved performance, even when solely focusing on the task of outlier removal [9, 25]. Given its important role in dealing with noisy data, the Euclidean k -MEANSOUT problem has received lots of attention from both theoretical and practical points of view. A series of algorithms have been proposed for the problem, including heuristics [9], distributed algorithms [10, 39, 24, 23, 27], approximation algorithms [25, 36, 6, 15, 30, 41], and coresets-construction methods [17, 19, 28, 29].

A commonly used way for relaxing the Euclidean k -MEANSOUT problem is to assume that the upper bounds on the numbers of centers and outliers (i.e., k and m) are small constants. Under this assumption, several PTASs exist for the problem. Feldman and Schulman [19] showed that a coresets-based approach yields a $(1 + \varepsilon)$ -approximation algorithm running in $nd(k + m)^{O(k+m)} + (\varepsilon^{-1}k \log n)^{O(1)}$ time, where n denotes the size of the given point set. Bhattacharya et al. [7] later gave an outlier-to-outlier-free reduction, where they mapped an instance of k -MEANSOUT to an instance of the standard k -means problem. This incurs an arbitrarily small loss in the approximation ratio and a $(k + m)^{m\varepsilon^{-O(1)}}$ multiplicative overhead on the running time of the executed algorithm. Subsequently, Agrawal et al. [1] and Jaiswal and Kumar [32] gave different reductions that impose multiplicative overheads of $n^{O(1)}((k + m)\varepsilon^{-1})^{O(m)}$ and $((k + m)\varepsilon^{-1})^{O(m)}$ on the running time, respectively. When combined with the state-of-the-art approximation scheme running in $O(ndk + d(k\varepsilon^{-1})^{O(1)} + (k\varepsilon^{-1})^{O(k\varepsilon^{-1})})$ time for the Euclidean k -means problem [18], these reductions yield $(1 + \varepsilon)$ -approximation algorithms with running times exponentially dependent on k and m . When k and m are not fixed, PTASs for Euclidean k -MEANSOUT exist for the case where d is a constant and the upper-bound constraint on the number of centers can be slightly violated, including the $(d\varepsilon^{-1})^{O(d)}$ -swap local-search algorithm given by Friggstad et al. [20] and the algorithm based on split-tree decomposition given by Cohen-Addad et al. [12]. These existing $(1 + \varepsilon)$ -approximation results are summarized in Table 1.

1.1 Our Results

As described above, there are $(1 + \varepsilon)$ -approximation algorithms for Euclidean k -MEANSOUT with running time exponential in both k and m . We cannot hope to achieve a better solution (i.e., an optimal one) in the same time frame: Euclidean k -MEANSOUT has been shown to be NP-hard, even for the case where $k = 2$ and $m = 0$ [38]. Nevertheless, this negative result does not rule out the possibility of achieving a $(1 + \varepsilon)$ -approximation solution in a more efficient manner. In particular, given that the outliers often constitute a constant fraction of the entire point set [21, 22, 15], it is interesting to consider whether a $(1 + \varepsilon)$ -approximation algorithm without exponential dependence on m exists in high-dimensional spaces. The main result in this paper is the first affirmative answer to this question, as described in Theorem 2.

■ **Table 1** $(1 + \varepsilon)$ -approximation algorithms for Euclidean k -MEANSOUT. The first two are bi-criteria approximation algorithms that violate the upper-bound constraint on the number of centers by a factor of $1 + O(\varepsilon)$. $T(n, d, k, \varepsilon) = O(ndk + d(k\varepsilon^{-1})^{O(1)} + (k\varepsilon^{-1})^{O(k\varepsilon^{-1})})$ denotes the running time of the state-of-the-art approximation scheme for the Euclidean k -means problem.

Running time	Parameter(s) in the exponent	Reference
$(nk)^{(d\varepsilon^{-1})^{O(d)}}$	d	[20]
$2^{\varepsilon^{-O(d^2)}} n \log^{O(1)} n + n^{O(1)}$	d	[12]
$nd(k+m)^{O(k+m)} + (\varepsilon^{-1}k \log n)^{O(1)}$	k, m	[19]
$(k+m)^{m\varepsilon^{-O(1)}} T(n, d, k, \varepsilon)$	k, m	[7]
$n^{O(1)} ((k+m)\varepsilon^{-1})^{O(m)} T(n, d, k, \varepsilon)$	k, m	[1]
$((k+m)\varepsilon^{-1})^{O(m)} T(n, d, k, \varepsilon)$	k, m	[32]
$nd((k+m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$	k	This work

► **Theorem 2.** *Given a constant $\varepsilon \in (0, 1)$ and an instance (\mathcal{P}, k, m) of Euclidean k -MEANSOUT satisfying $\mathcal{P} \subset \mathbb{R}^d$ and $|\mathcal{P}| = n$, there is a $(1 + \varepsilon)$ -approximation algorithm running in $n^2 d ((k+m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ time.*

Leveraging a coresets construction method that reduces the point set to a weighted set of size $\text{poly}(m, k, \varepsilon^{-1})$ [29], we can improve the running time of the algorithm in Theorem 2 to be linear in n . A detailed analysis is given in Section 4.2.

Awasthi et al. [3] showed that obtaining a PTAS for Euclidean k -MEANSOUT for arbitrary k and $d = \Omega(\log n)$ is also NP-hard. Given that we avoid the exponential dependence on d , the exponential dependence on k exhibited in Theorem 2 is unavoidable. Existing approximation schemes without exponential dependence on k , like the ones in [20, 12], work only in low-dimensional spaces and select more than k centers.

The optimal solutions to instances of k -MEANSOUT exhibit a useful property: The center associated with each point is simply the one nearest to the point. This property, called the *locality* property, guides the estimation of the locations of centers selected by an optimal solution. One advantage of our approach is that it no longer relies on the locality property. This enhances the versatility of the approach, allowing us to deal with problems not satisfying the locality property. Indeed, we show that our approach establishes a unified framework for addressing generalizations of Euclidean k -MEANSOUT that invalidate the locality property, including the Euclidean versions of *capacitated* and *fair* k -MEANSOUT [32, 13]. As in the unconstrained case, we give the first PTAS for each considered generalization of Euclidean k -MEANSOUT, assuming k is a constant.

1.2 Our Techniques

Most existing approximation schemes for k -MEANSOUT are built heavily on the following natural idea: The m outliers can be viewed as m virtual centers, each corresponding to a cluster containing only itself, and solving a $(k+m)$ -clustering problem enables the identification of the k centers and m outliers. This provides a clear strategy for constructing the desired approximation solution. However, the complexities of clustering problems increase with the number of centers to be identified, and considering the additional m virtual centers incurs an exponential time-dependence on m , as exhibited in the running times of the approximation schemes given in [19, 7, 1, 32]. To deal with the case where m is super-constant, we employ a different sampling-based method.

Given a small positive constant ε , it is well-known that the centroid of $O(\varepsilon^{-1})$ uniformly sampled points is close to the optimal 1-means clustering center of the entire point set (Lemma 6). Building upon this insight, we uniformly sample from the point set for each cluster defined by an optimal solution, and enumerate the sampled points to find a subset of $O(\varepsilon^{-1})$ points uniformly distributed in the cluster, such that the corresponding center can be approximated by the centroid of the subset. This idea follows the one for the k -means problem outlined in [37], while the case we consider poses more challenges. The first issue we encounter lies in the presence of a non-fixed number of outliers, which reduces the proportion of some small clusters within the point set, and so, the likelihood of obtaining members of these clusters through randomly sampling may not be sufficiently high. To address this issue, we carefully adjust the sampling region for each cluster in a recursive way, ensuring a sufficient proportion of the cluster within the defined region. Another issue emerges as we extend our consideration to the constrained variants of k -MEANSOUT that do not satisfy the locality property. In these variants, the points are not guaranteed to be close to their corresponding centers. This leads to the lack of a distinct pattern in the distributions of the clusters, making it more difficult to determine appropriate sampling regions. In dealing with this issue, it is essential to address the points that violate the locality property, meaning those far from their corresponding centers. Instead of attempting to find these points through sampling, we regard previously identified approximate centers close to these points as substitutes, enumerating the union of the sampled points and the previously selected centers to construct a small representative set for the considered cluster.

2 Preliminaries

Given a positive integer λ , define $[\lambda] = \{1, \dots, \lambda\}$. Given a set $\mathcal{X} \subset \mathbb{R}^d$ and a point $y \in \mathbb{R}^d$, let $\Delta(y, \mathcal{X}) = \min_{x \in \mathcal{X}} \|y - x\|^2$ denote the squared distance from y to the nearest point in \mathcal{X} , and let $\Delta(\mathcal{X}, y) = \sum_{x \in \mathcal{X}} \|x - y\|^2$ denote the sum of squared distances from y to the points in \mathcal{X} . Additionally, define $c(\mathcal{X}) = |\mathcal{X}|^{-1} \sum_{x \in \mathcal{X}} x$ as the centroid of \mathcal{X} , and let $\Delta(\mathcal{X}) = \min_{c \in \mathbb{R}^d} \Delta(\mathcal{X}, c)$ denote the minimum 1-means clustering cost of \mathcal{X} .

The following two lemmas provide ways of estimating the squared distances from the points to the centers selected by an approximate solution. As a corollary of the first one, we know that $\Delta(\mathcal{X}) = \Delta(\mathcal{X}, c(\mathcal{X}))$ for each $\mathcal{X} \subset \mathbb{R}^d$.

► **Lemma 3** ([35]). *Given a point $x \in \mathbb{R}^d$ and a set $\mathcal{X} \subset \mathbb{R}^d$, we have $\Delta(\mathcal{X}, x) = \Delta(\mathcal{X}) + |\mathcal{X}| \cdot \|c(\mathcal{X}) - x\|^2$.*

► **Lemma 4** ([16]). *Given a set $\mathcal{X} \subset \mathbb{R}^d$, a real number $\lambda \in (0, 1]$, and a subset $\mathcal{X}' \subseteq \mathcal{X}$ satisfying $|\mathcal{X}'| \geq \lambda|\mathcal{X}|$, we have $\|c(\mathcal{X}') - c(\mathcal{X})\|^2 \leq (1 - \lambda)(\lambda|\mathcal{X}|)^{-1} \Delta(\mathcal{X})$.*

The following lemma is an extensive version of triangle inequality.

► **Lemma 5.** *Given three points x, y , and z in \mathbb{R}^d and a real number $\lambda > 0$, we have $\|x - z\|^2 \leq (1 + \lambda)\|x - y\|^2 + (1 + \lambda^{-1})\|y - z\|^2$.*

Proof. Using triangle inequality, we have $\|x - z\| \leq \|x - y\| + \|y - z\|$, which implies that

$$\begin{aligned} \|x - z\|^2 &\leq (\|x - y\| + \|y - z\|)^2 \\ &= \|x - y\|^2 + \|y - z\|^2 + 2\sqrt{\lambda}\|x - y\| \frac{1}{\sqrt{\lambda}}\|y - z\| \\ &\leq \|x - y\|^2 + \|y - z\|^2 + \lambda\|x - y\|^2 + \frac{1}{\lambda}\|y - z\|^2, \end{aligned}$$

as desired. ◀

The following result says that uniform sampling works for the 1-means problem.

► **Lemma 6** ([31]). *Given a set $\mathcal{X} \subset \mathbb{R}^d$, a multi-set \mathcal{S} constructed by sampling points from \mathcal{X} independently and uniformly, and a positive real number λ , inequality $\|c(\mathcal{S}) - c(\mathcal{X})\|^2 \leq (\lambda|\mathcal{S}||\mathcal{X}|)^{-1}\Delta(\mathcal{X})$ holds with probability at least $1 - \lambda$.*

The following result is known as Chernoff bound, which has been widely used in analysis of sampling-based algorithms.

► **Lemma 7** ([26]). *Given a set of t independent random variables a_1, \dots, a_t and a real number $p \in (0, 1)$, if $a_i \in \{0, 1\}$ and $\Pr[a_i = 1] \geq p$ hold for each $i \in [t]$, then each real number $\lambda \in (0, 1)$ satisfies $\Pr\left[\sum_{i=1}^t a_i < (1 - \lambda)pt\right] < e^{-\frac{1}{2}\lambda^2 pt}$.*

As a corollary of Lemma 7, we have the following result about uniform sampling.

► **Lemma 8.** *Given a set $\mathcal{X} \subset \mathbb{R}^d$, a subset $\mathcal{S} \subseteq \mathcal{X}$, a positive integer t , and a real number $\lambda \in (0, 1)$, the following event happens with probability more than $1 - e^{-\frac{\lambda^2 t |\mathcal{S}|}{2|\mathcal{X}|}}$: A multi-set of more than t points independently and uniformly sampled from \mathcal{X} contains no less than $(1 - \lambda)t|\mathcal{S}||\mathcal{X}|^{-1}$ points in \mathcal{S} .*

Proof. We define a set of independent random variables a_1, \dots, a_t as follows: For each $i \in [t]$, let $a_i = 1$ if the i -th point sampled from \mathcal{X} is in \mathcal{S} , and let $a_i = 0$ otherwise. We have $\Pr[a_i = 1] = |\mathcal{S}||\mathcal{X}|^{-1}$ for each $i \in [t]$. Lemma 7 implies that

$$\Pr\left[\sum_{i=1}^t a_i \geq (1 - \lambda)t|\mathcal{S}||\mathcal{X}|^{-1}\right] = 1 - \Pr\left[\sum_{i=1}^t a_i < (1 - \lambda)t|\mathcal{S}||\mathcal{X}|^{-1}\right] > 1 - e^{-\frac{\lambda^2 t |\mathcal{S}|}{2|\mathcal{X}|}}.$$

This completes the proof of Lemma 8. ◀

3 The Sampling Algorithm

In this section we give a sampling-based approach for constructing candidate center sets, as described in Algorithm 1. Taking as inputs three real numbers k , m , and ε , two sets \mathcal{C}' and \mathcal{P}^\dagger , and a collection \mathbb{C} , the algorithm recursively augments \mathbb{C} with some center sets. Here, k is the upper bound on the size of a center set, m is the upper bound on the number of outliers, ε is the factor trading off the approximation ratio and running time, \mathcal{C}' is a center set that needs to be updated or added to \mathbb{C} , \mathcal{P}^\dagger is the sampling region, and \mathbb{C} contains the center sets that have been constructed. The algorithm constructs a multi-set \mathcal{S} as follows: It independently and uniformly samples $O((k + m)\varepsilon^{-3})$ points from \mathcal{P}^\dagger and then adds to the set $O(\varepsilon^{-1})$ copies of each center in \mathcal{C}' . After constructing \mathcal{S} , the algorithm considers each subset of size $O(\varepsilon^{-1})$ of \mathcal{S} , adding the centroid of the subset to \mathcal{C}' and recursively invoking itself with the updated center set. Finally, it throws away half of the points in \mathcal{P}^\dagger that are close to the centers in \mathcal{C}' , and invokes itself again with the reduced point set.

By inducting on the sizes of the given sets of centers and points, we obtain the following upper bounds on the running time of our sampling algorithm and the quantity of center sets it generates.

► **Lemma 9.** *Given a constant $\varepsilon \in (0, 1)$, a collection \mathbb{C} , and an instance (\mathcal{P}, k, m) of Euclidean k -MEANSOUT with $|\mathcal{P}| \leq n$ and $\mathcal{P} \subset \mathbb{R}^d$, **Sampling** $(k, m, \varepsilon, \emptyset, \mathcal{P}, \mathbb{C})$ runs in $nd((k + m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ time and adds at most $n((k + m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ center sets to \mathbb{C} .*

Algorithm 1 $\text{Sampling}(k, m, \varepsilon, \mathcal{C}', \mathcal{P}^\dagger, \mathbb{C})$.

Input: Two positive integers k and m , a constant $\varepsilon \in (0, 1)$, a set $\mathcal{C}' \subset \mathbb{R}^d$ of no more than k centers, a set $\mathcal{P}^\dagger \subset \mathbb{R}^d$ of points, and a collection \mathbb{C} of center sets

- 1 $N \leftarrow \lceil (17400k + 60m)\varepsilon^{-3} \rceil$, $M \leftarrow \lceil 25\varepsilon^{-1} \rceil$;
- 2 **if** $|\mathcal{C}'| = k$ **then**
- 3 $\mathbb{C} \leftarrow \mathbb{C} \cup \{\mathcal{C}'\}$;
- 4 **else**
- 5 Sample a multi-set \mathcal{S} of N points from \mathcal{P}^\dagger independently and uniformly;
- 6 $\mathcal{S} \leftarrow \mathcal{S} \uplus \{M \text{ copies of each } c \in \mathcal{C}'\}$;
- 7 **for each** $\mathcal{S}' \subset \mathcal{S}$ **satisfying** $|\mathcal{S}'| = M$ **do**
- 8 Calculate the centroid $c(\mathcal{S}')$ of \mathcal{S}' ;
- 9 $\text{Sampling}(k, m, \varepsilon, \mathcal{C}' \uplus \{c(\mathcal{S}')\}, \mathcal{P}^\dagger, \mathbb{C})$;
- 10 **if** $\mathcal{C}' \neq \emptyset$ **and** $|\mathcal{P}^\dagger| > 1$ **then**
- 11 Let \mathcal{P}^\ddagger be the set of the $\lfloor \frac{|\mathcal{P}^\dagger|}{2} \rfloor$ points $p \in \mathcal{P}^\dagger$ with the largest values of $\Delta(p, \mathcal{C}')$;
- 12 $\text{Sampling}(k, m, \varepsilon, \mathcal{C}', \mathcal{P}^\ddagger, \mathbb{C})$.

3.1 An Overview of Analysis

We now introduce some notations to be used throughout this section. We consider a constant $\varepsilon \in (0, 1)$ and an instance $\mathcal{I} = (\mathcal{P}, k, m)$ of Euclidean k -MEANSOUT, where $\mathcal{P} \subset \mathbb{R}^d$ and $|\mathcal{P}| = n$. Let $N = \lceil (17400k + 60m)\varepsilon^{-3} \rceil$ and $M = \lceil 25\varepsilon^{-1} \rceil$. Let $\mathcal{P}_1, \dots, \mathcal{P}_k, \mathcal{O}$ denote $k + 1$ arbitrary disjoint subsets of \mathcal{P} satisfying $|\mathcal{O}| = m$, $\bigcup_{i=1}^k \mathcal{P}_i \cup \mathcal{O} = \mathcal{P}$, and $|\mathcal{P}_1| \geq |\mathcal{P}_2| \geq \dots \geq |\mathcal{P}_k|$. For each $i \in [k]$, let $c_i^* = c(\mathcal{P}_i)$ be the centroid of \mathcal{P}_i . Define $\Delta(\mathbb{P}) = \sum_{i=1}^k \Delta(\mathcal{P}_i)$. Let \mathbb{C} denote the collection of center sets constructed by $\text{Sampling}(k, m, \varepsilon, \emptyset, \mathcal{P}, \emptyset)$. We will show that \mathbb{C} contains a center set approximating $\{c_1^*, \dots, c_k^*\}$ well with high probability. More formally, we will prove the correctness of the following result.

► **Lemma 10.** *The following event happens with probability no less than 15^{-k} : There is a center set $\mathcal{C} \in \mathbb{C}$ satisfying $\sum_{i=1}^k \min_{c \in \mathcal{C}} \Delta(\mathcal{P}_i, c) \leq (1 + \varepsilon)\Delta(\mathbb{P})$.*

The proof of Lemma 10, presented in Section 3.2, is based on an inductive method. Specifically, for a given integer $i \in \{2, \dots, k\}$, we assume that a set $\mathcal{C}_{i-1} = \{c_1, \dots, c_{i-1}\}$ of centers, where c_j is close to c_j^* for each $j \in [i-1]$, has been constructed, and prove that a center close to c_i^* can be identified and added to \mathcal{C}_{i-1} when invoking Algorithm 1 with \mathcal{C}_{i-1} . Define (informally) \mathcal{B}_{i-1} as the set of points close to one of the centers in \mathcal{C}_{i-1} . Given a real number λ defined based on the value of ε , we divide the analysis into the following two cases: (1) $|\mathcal{P}_i \setminus \mathcal{B}_{i-1}| \leq \lambda |\mathcal{P}_i|$, and (2) $|\mathcal{P}_i \setminus \mathcal{B}_{i-1}| > \lambda |\mathcal{P}_i|$.

The case of $|\mathcal{P}_i \setminus \mathcal{B}_{i-1}| \leq \lambda |\mathcal{P}_i|$ captures the scenario where most points in \mathcal{P}_i are from \mathcal{B}_{i-1} and in close proximity to one of the centers in \mathcal{C}_{i-1} . Conditioned on this, the centers copied in step 6 of Algorithm 1 can be regarded as proxies for the points in \mathcal{P}_i . Based on Lemma 4 and Lemma 6, we are able to show that the centroid of a subset of the copies is close to c_i^* and can be added to \mathcal{C}_{i-1} by the algorithm.

For the case where $|\mathcal{P}_i \setminus \mathcal{B}_{i-1}| > \lambda |\mathcal{P}_i|$, we handle the points from $\mathcal{P}_i \cap \mathcal{B}_{i-1}$ similarly to the above approach: We regard the copies of centers in \mathcal{C}_{i-1} as proxies of these points. It remains to consider how to deal with the points from $\mathcal{P}_i \setminus \mathcal{B}_{i-1}$. When formally defining \mathcal{B}_{i-1} in Section 3.2, we carefully establish its range such that the ratio of $|\mathcal{P} \setminus \mathcal{B}_{i-1}|$ to $|\mathcal{P}_i \setminus \mathcal{B}_{i-1}|$ is

polynomial in m , k , and ε if $|\mathcal{P}_i \setminus \mathcal{B}_{i-1}| > \lambda |\mathcal{P}_i|$ (as exhibited in Claim 15), which implies that a limited number of points uniformly sampled from $\mathcal{P} \setminus \mathcal{B}_{i-1}$ contains a representative subset of $\mathcal{P}_i \setminus \mathcal{B}_{i-1}$ with a good chance. Furthermore, it is shown that Algorithm 1 can recursively adjust the sampling region to make it close to $\mathcal{P} \setminus \mathcal{B}_{i-1}$, based on the operation in step 12. Putting everything together, we know that a representative subset of $\mathcal{P}_i \setminus \mathcal{B}_{i-1}$ is involved in the points sampled by the algorithm.

3.2 Proof of Lemma 10

It can be seen that the algorithm **Sampling** makes multiple recursive calls to itself. We conceptualize the execution of **Sampling**($k, m, \varepsilon, \emptyset, \mathcal{P}, \emptyset$) as a tree denoted by \mathcal{T} . Each node within the tree, identified by $(\mathcal{C}', \mathcal{P}^\dagger)$, corresponds to an invocation of the algorithm with center set \mathcal{C}' and sampling region \mathcal{P}^\dagger . The children of a node symbolize the recursive calls made in the corresponding invocation of the algorithm, and each leaf of the tree is associated with a set of k centers added to \mathbb{C} .

Prior to showing the correctness of Lemma 10, we establish the following invariant for each $i \in [k]$.

$\tau(i)$: With probability at least 15^{-i} , there exists a node $(\mathcal{C}_i, \mathcal{P}^\dagger)$ in \mathcal{T} such that (1) \mathcal{C}_i consists of i centers c_1, \dots, c_i (added in this order), (2) each $j \in [i]$ satisfies $\Delta(\mathcal{P}_j, c_j) \leq (1 + \frac{\varepsilon}{2})\Delta(\mathcal{P}_j) + \frac{\varepsilon}{2k}\Delta(\mathbb{P})$, and (3) $\{p \in \mathcal{P} : \Delta(p, \mathcal{C}_i) > \frac{\varepsilon\Delta(\mathbb{P})}{8k|\mathcal{P}_i|}\} \subseteq \mathcal{P}^\dagger$.

We prove the invariant by induction on i . We first consider the base case of $i = 1$. It can be seen that **Sampling**($k, m, \varepsilon, \emptyset, \mathcal{P}, \emptyset$) independently and uniformly samples a multi-set \mathcal{S} of N points from \mathcal{P} , and \mathcal{T} has a node $(\{c(\mathcal{S}'), \mathcal{P}\})$ for each $\mathcal{S}' \subseteq \mathcal{S}$ with $|\mathcal{S}'| = M$. We have $|\mathcal{P}||\mathcal{P}_1|^{-1} = (\sum_{j=1}^k |\mathcal{P}_j| + |\mathcal{O}|)|\mathcal{P}_1|^{-1} \leq k + m$ due to the fact that $|\mathcal{P}_1| \geq |\mathcal{P}_2| \geq \dots \geq |\mathcal{P}_k|$. This inequality and Lemma 8 (with $t = N$ and $\lambda = 1 - M(k + m)N^{-1}$) imply that a node $(\{c(\mathcal{S}'), \mathcal{P}\})$ satisfying $|\mathcal{S}'| = M$ and $\mathcal{S}' \subseteq \mathcal{P}_1$ exists in \mathcal{T} with probability more than $1 - e^{-(1 - M(k + m)N^{-1})^2 N / (2(k + m))} > \frac{1}{3}$. Using Lemma 6 (with $\lambda = \frac{4}{5}$), we know that if such a node $(\{c(\mathcal{S}'), \mathcal{P}\})$ exists, then inequality

$$\|c(\mathcal{S}') - c_1^*\|^2 \leq \frac{5\Delta(\mathcal{P}_1)}{4|\mathcal{P}_1||\mathcal{S}'|} = \frac{\varepsilon\Delta(\mathcal{P}_1)}{20|\mathcal{P}_1|} \quad (1)$$

holds with probability at least $\frac{1}{5}$. Inequality (1) and Lemma 3 imply that

$$\Delta(\mathcal{P}_1, c(\mathcal{S}')) = \Delta(\mathcal{P}_1) + |\mathcal{P}_1| \cdot \|c(\mathcal{S}') - c_1^*\|^2 \leq (1 + \frac{\varepsilon}{20})\Delta(\mathcal{P}_1),$$

which in turn implies that $\tau(1)$ is true.

We now assume that $\tau(i-1)$ holds for an integer $i \in \{2, \dots, k\}$, and prove that $\tau(i)$ also holds. Let $(\mathcal{C}_{i-1}, \mathcal{P}^\dagger)$ be a node satisfying

$$\Delta(\mathcal{P}_j, c_j) \leq (1 + \frac{\varepsilon}{2})\Delta(\mathcal{P}_j) + \frac{\varepsilon}{2k}\Delta(\mathbb{P}) \quad (2)$$

for each $j \in [i-1]$ and

$$\{p \in \mathcal{P} : \Delta(p, \mathcal{C}_{i-1}) > \frac{\varepsilon\Delta(\mathbb{P})}{8k|\mathcal{P}_{i-1}|}\} \subseteq \mathcal{P}^\dagger, \quad (3)$$

where $\mathcal{C}_{i-1} = \{c_1, \dots, c_{i-1}\}$. $\tau(i-1)$ implies that such a node $(\mathcal{C}_{i-1}, \mathcal{P}^\dagger)$ exists in \mathcal{T} with probability no less than 15^{1-i} . Conditioning on the existence of this node, we define $\mathcal{B}_{i-1} = \{p \in \mathcal{P} : \Delta(p, \mathcal{C}_{i-1}) \leq \frac{\varepsilon\Delta(\mathbb{P})}{8k|\mathcal{P}_i|}\}$. Let $\mathcal{P}_i^n = \mathcal{P}_i \cap \mathcal{B}_{i-1}$ and $\mathcal{P}_i^f = \mathcal{P}_i \setminus \mathcal{B}_{i-1}$ for brevity. As described in Section 3.1, we prove $\tau(i)$ differently based on the size of \mathcal{P}_i^f . Specifically, we consider the following two cases: (1) $|\mathcal{P}_i^f| \leq \frac{\varepsilon}{17}|\mathcal{P}_i|$, and (2) $|\mathcal{P}_i^f| > \frac{\varepsilon}{17}|\mathcal{P}_i|$. These two cases are respectively analyzed in the following two subsections.

Case (1): $|\mathcal{P}_i^f| \leq \frac{\varepsilon}{17} |\mathcal{P}_i|$

In this case, most points in \mathcal{P}_i are close to the centers in \mathcal{C}_{i-1} , based on which we show that a convex combination of the latter's members effectively approximates c_i^* . We consider a multi-set $\mathcal{P}'_i = \{c(p) : p \in \mathcal{P}_i^n\}$, where $c(p)$ is the center in \mathcal{C}_{i-1} nearest to p . The proximity of each point in \mathcal{P}_i^n to its counterpart in \mathcal{P}'_i , combined with the substantial proportion of \mathcal{P}_i^n in \mathcal{P}_i , implies that the centroid of \mathcal{P}'_i is close to c_i^* . This is confirmed by the following lemma.

► **Lemma 11.** *If $|\mathcal{P}_i^f| \leq \frac{\varepsilon}{17} |\mathcal{P}_i|$, then we have $\|c(\mathcal{P}'_i) - c_i^*\|^2 \leq \frac{\varepsilon \Delta(\mathcal{P}_i)}{8|\mathcal{P}_i|} + \frac{\varepsilon \Delta(\mathbb{P})}{4k|\mathcal{P}_i|}$.*

Lemma 11 suggests that $c(\mathcal{P}'_i)$ is close to c_i^* . Unfortunately, directly approximating c_i^* using $c(\mathcal{P}'_i)$ is not feasible, as the members of both \mathcal{P}_i^n and \mathcal{P}'_i are unknown. The idea of Algorithm 1 is to take M copies of each center from \mathcal{C}_{i-1} and simulate \mathcal{P}'_i using a subset of these copies. The following lemma implies that this yields a center approximating c_i^* well with high probability and, furthermore, generates the node claimed in $\tau(i)$.

► **Lemma 12.** *If $|\mathcal{P}_i^f| \leq \frac{\varepsilon}{17} |\mathcal{P}_i|$, then the following event happens with probability at least $\frac{1}{5}$: $(\mathcal{C}_{i-1}, \mathcal{P}^\dagger)$ has a child $(\mathcal{C}_{i-1} \uplus \{c_i\}, \mathcal{P}^\dagger)$ satisfying $\{p \in \mathcal{P} : \Delta(p, \mathcal{C}_{i-1} \uplus \{c_i\}) > \frac{\varepsilon \Delta(\mathbb{P})}{8k|\mathcal{P}_i|}\} \subseteq \mathcal{P}^\dagger$ and $\Delta(\mathcal{P}_i, c_i) < (1 + \frac{\varepsilon}{2})\Delta(\mathcal{P}_i) + \frac{\varepsilon}{2k}\Delta(\mathbb{P})$.*

Proof. The fact that $|\mathcal{P}_1| \geq |\mathcal{P}_2| \geq \dots \geq |\mathcal{P}_k|$ and $\Delta(p, \mathcal{C}_{i-1}) \geq \Delta(p, \mathcal{C}_{i-1} \uplus \{c\})$ for each $p \in \mathcal{P}$ and $c \in \mathbb{R}^d$ suggests that

$$\{p \in \mathcal{P} : \Delta(p, \mathcal{C}_{i-1} \uplus \{c\}) > \frac{\varepsilon \Delta(\mathbb{P})}{8k|\mathcal{P}_i|}\} \subseteq \{p \in \mathcal{P} : \Delta(p, \mathcal{C}_{i-1}) > \frac{\varepsilon \Delta(\mathbb{P})}{8k|\mathcal{P}_{i-1}|}\} \subseteq \mathcal{P}^\dagger \quad (4)$$

for each $c \in \mathbb{R}^d$, where the last step is due to inequality (3).

In the invocation of Algorithm 1 corresponding to $(\mathcal{C}_{i-1}, \mathcal{P}^\dagger)$, the algorithm takes M copies of each center from \mathcal{C}_{i-1} and calculates the centroid of each subset of the copies with size M . By Lemma 6 (with $\lambda = \frac{4}{5}$) and the fact that $\mathcal{P}'_i \subseteq \mathcal{C}_{i-1}$, we know that a center c_i identified in this way satisfies

$$\|c_i - c(\mathcal{P}'_i)\|^2 \leq \frac{5\Delta(\mathcal{P}'_i)}{4M|\mathcal{P}'_i|} \leq \frac{\varepsilon \Delta(\mathcal{P}'_i)}{20|\mathcal{P}'_i|} = \frac{\varepsilon \Delta(\mathcal{P}'_i)}{20|\mathcal{P}_i^n|} \quad (5)$$

with probability no less than $\frac{1}{5}$.

Denote by c_i a center satisfying inequality (5). Intuitively, inequality (5) and Lemma 11 imply an upper bound on the squared distance from c_i to c_i^* . Combining this insight with Lemma 3, we can derive the following claim.

► **Claim 13.** *If $|\mathcal{P}_i^f| \leq \frac{\varepsilon}{17} |\mathcal{P}_i|$, then we have $\Delta(\mathcal{P}_i, c_i) < (1 + \frac{\varepsilon}{2})\Delta(\mathcal{P}_i) + \frac{\varepsilon}{2k}\Delta(\mathbb{P})$.*

Using Claim 13 and inequality (4), we complete the proof of Lemma 12. ◀

Case (2): $|\mathcal{P}_i^f| > \frac{\varepsilon}{17} |\mathcal{P}_i|$

As in the previous case, we simulate the points in \mathcal{P}_i^n using the centers in \mathcal{C}_{i-1} . The main challenge in the current case is that we cannot ignore the points from \mathcal{P}_i^f as we did previously, since their proportion in \mathcal{P}_i is no longer bounded by a small value. As a remedy, we argue that we can sample sufficient points from \mathcal{P}_i^f in step 5 of Algorithm 1 by recursively adjusting the sampling region. Furthermore, we will show that a combination of these sampled points and the centers in \mathcal{C}_{i-1} approximates c_i^* well.

The following lemma suggests a lower bound on the proportion of \mathcal{P}_i^f within a carefully selected sampling region.

► **Lemma 14.** *If $|\mathcal{P}_i^f| > \frac{\varepsilon}{17}|\mathcal{P}_i|$, then there is a node $(\mathcal{C}_{i-1}, \mathcal{P}^\ddagger)$ in the descendants of $(\mathcal{C}_{i-1}, \mathcal{P}^\dagger)$ (including itself) that satisfies $\mathcal{P} \setminus \mathcal{B}_{i-1} \subseteq \mathcal{P}^\ddagger$ and $\varepsilon^{-2}(580k + 2m)|\mathcal{P}_i^f| > |\mathcal{P}^\ddagger|$.*

Proof. Our idea for proving Lemma 14 is to show that $|\mathcal{P}_i^f|$ is not too small compared to $|\mathcal{P} \setminus \mathcal{B}_{i-1}|$, and then argue that the invocation of Algorithm 1 corresponding to $(\mathcal{C}_{i-1}, \mathcal{P}^\dagger)$ can find a sampling region \mathcal{P}^\ddagger close to $\mathcal{P} \setminus \mathcal{B}_{i-1}$.

The following claim establishes a lower bound on the ratio of $|\mathcal{P}_i^f|$ to $|\mathcal{P} \setminus \mathcal{B}_{i-1}|$.

▷ **Claim 15.** *If $|\mathcal{P}_i^f| > \frac{\varepsilon}{17}|\mathcal{P}_i|$, then $\varepsilon^{-2}(290k + m)|\mathcal{P}_i^f| > |\mathcal{P} \setminus \mathcal{B}_{i-1}|$.*

The fact that $|\mathcal{P}_1| \geq |\mathcal{P}_2| \geq \dots \geq |\mathcal{P}_k|$ and inequality (3) imply that

$$\mathcal{P} \setminus \mathcal{B}_{i-1} = \{p \in \mathcal{P} : \Delta(p, \mathcal{C}_{i-1}) > \frac{\varepsilon \Delta(\mathbb{P})}{8k|\mathcal{P}_i|}\} \subseteq \mathcal{P}^\dagger. \quad (6)$$

We sort the points $p \in \mathcal{P}^\dagger$ by decreasing values of $\Delta(p, \mathcal{C}_{i-1})$. Let p_t be the t -th point in this order for each $t \in [|\mathcal{P}^\dagger|]$, and define $\mathcal{P}_s^\dagger = \{p_t : t \in [2^{-s}|\mathcal{P}^\dagger|]\}$ for each integer $s \in [0, \lceil \log |\mathcal{P}^\dagger| \rceil]$. Equality (6) implies the existence of an integer $\tilde{s} \in [0, \lceil \log |\mathcal{P}^\dagger| \rceil]$ satisfying $\mathcal{P} \setminus \mathcal{B}_{i-1} \subseteq \mathcal{P}_{\tilde{s}}^\dagger$ and $2|\mathcal{P} \setminus \mathcal{B}_{i-1}| \geq |\mathcal{P}_{\tilde{s}}^\dagger|$, and we have $\varepsilon^{-2}(580k + 2m)|\mathcal{P}_i^f| > |\mathcal{P}_{\tilde{s}}^\dagger|$ due to Claim 15. Moreover, the operations performed in steps 12 and 13 of Algorithm 1 ensure that $(\mathcal{C}_{i-1}, \mathcal{P}_{\tilde{s}}^\dagger)$ is a descendant of $(\mathcal{C}_{i-1}, \mathcal{P}^\dagger)$. This completes the proof of Lemma 14. ◀

Following the approach in Case (1), we consider a multi-set where each $p \in \mathcal{P}_i^n$ is replaced by the nearest center $c(p)$ in \mathcal{C}_{i-1} . We define the multi-set as $\mathcal{P}'_i = \{c(p) : p \in \mathcal{P}_i^n\} \cup \mathcal{P}_i^f$. Intuitively, we can closely simulate this multi-set using the union of a subset of $\{c(p) : p \in \mathcal{P}_i^n\}$ and a set of points sampled from \mathcal{P}_i^f , and the centroid of the simulated set is close to c_i^* , given that the squared distance from each $p \in \mathcal{P}_i^n$ to $c(p)$ is upper-bounded by a small value and sufficient points from \mathcal{P}_i^f can be sampled. This motivates the following lemma.

► **Lemma 16.** *If $|\mathcal{P}_i^f| > \frac{\varepsilon}{17}|\mathcal{P}_i|$, then the following event happens with probability at least $\frac{1}{15}$: $(\mathcal{C}_{i-1}, \mathcal{P}^\dagger)$ has a descendant $(\mathcal{C}_{i-1} \uplus \{c_i\}, \mathcal{P}^\ddagger)$ with $\{p \in \mathcal{P} : \Delta(p, \mathcal{C}_{i-1} \uplus \{c_i\}) > \frac{\varepsilon \Delta(\mathbb{P})}{8k|\mathcal{P}_i|}\} \subseteq \mathcal{P}^\ddagger$ and $\Delta(\mathcal{P}_i, c_i) < (1 + \frac{\varepsilon}{2})\Delta(\mathcal{P}_i) + \frac{\varepsilon}{2k}\Delta(\mathbb{P})$.*

Proof. Denote by $(\mathcal{C}_{i-1}, \mathcal{P}^\ddagger)$ the descendant of $(\mathcal{C}_{i-1}, \mathcal{P}^\dagger)$ claimed in Lemma 14. When calling Algorithm 1 with $(\mathcal{C}_{i-1}, \mathcal{P}^\ddagger)$, we independently and uniformly sample N points from \mathcal{P}^\ddagger and take M copies of each center from \mathcal{C}_{i-1} to construct a multi-set \mathcal{S} . $(\mathcal{C}_{i-1}, \mathcal{P}^\ddagger)$ has a child $(\mathcal{C}_{i-1} \uplus \{c(\mathcal{S}')\}, \mathcal{P}^\ddagger)$ for each $\mathcal{S}' \subset \mathcal{S}$ with $|\mathcal{S}'| = M$. By Lemma 8 (with $t = N$ and $\lambda = \frac{1}{6}$) and the fact that $\mathcal{P}'_i \subseteq \mathcal{P} \setminus \mathcal{B}_{i-1} \subseteq \mathcal{P}^\ddagger$ and $\varepsilon^{-2}(580k + 2m)|\mathcal{P}_i^f| > |\mathcal{P}^\ddagger|$ (due to Lemma 14), we know that \mathcal{S} contains no less than M points uniformly distributed in \mathcal{P}'_i with probability at least $1 - e^{-\varepsilon^2 N / (72(580k + 2m))} > \frac{1}{3}$. Moreover, the probability that \mathcal{S} has a subset \mathcal{S}' consisting of M points uniformly distributed in \mathcal{P}'_i can be lower-bounded by the same constant, given that there are M copies of each distinct member of $\mathcal{P}'_i \setminus \mathcal{P}_i^f$ within \mathcal{S} . Under the assumption that such a subset \mathcal{S}' exists, Lemma 4 (with $\lambda = \frac{4}{5}$) implies that inequality

$$\|c_i - c(\mathcal{P}'_i)\|^2 \leq \frac{5\Delta(\mathcal{P}'_i)}{4M|\mathcal{P}'_i|} = \frac{\varepsilon\Delta(\mathcal{P}'_i)}{20|\mathcal{P}_i|} \quad (7)$$

holds with probability at least $\frac{1}{5}$, where $c_i = c(\mathcal{S}')$ is the centroid of \mathcal{S}' . Putting everything together, we know that $(\mathcal{C}_{i-1}, \mathcal{P}^\ddagger)$ has a child $(\mathcal{C}_{i-1} \uplus \{c_i\}, \mathcal{P}^\ddagger)$ satisfying inequality (7) with probability at least $\frac{1}{15}$.

We now show that $(\mathcal{C}_{i-1} \uplus \{c_i\}, \mathcal{P}^\ddagger)$ satisfies the properties claimed in Lemma 16. By a similar argument as in the proof of Claim 13, we can obtain the following upper bound on $\Delta(\mathcal{P}_i, c_i)$.

■ **Algorithm 2** The Algorithm for Euclidean k -MEANSOUT.

Input: A constant $\varepsilon \in (0, 1)$ and an instance (\mathcal{P}, k, m) of Euclidean k -MEANSOUT satisfying $\mathcal{P} \subset \mathbb{R}^d$

Output: A set $\mathcal{C} \subset \mathbb{R}^d$ of no more than k centers and a set $\mathcal{O} \subseteq \mathcal{P}$ of no more than m outliers

- 1 $\mathbb{C} \leftarrow \emptyset$;
- 2 **for** $t \leftarrow 1$ **to** 15^k **do**
- 3 $\mathbb{C} \leftarrow \text{Sampling}(k, m, \varepsilon, \emptyset, \mathcal{P}, \mathbb{C})$;
- 4 **for** *each* $\mathcal{C}' \in \mathbb{C}$ **do**
- 5 $\text{cost}(\mathcal{C}') \leftarrow \min_{\mathcal{O}' \subseteq \mathcal{P} \wedge |\mathcal{O}'| \leq m} \sum_{p \in \mathcal{P} \setminus \mathcal{O}'} \Delta(p, \mathcal{C}')$;
- 6 $\mathcal{C} \leftarrow \arg \min_{\mathcal{C}' \in \mathbb{C}} \text{cost}(\mathcal{C}')$;
- 7 $\mathcal{O} \leftarrow \arg \max_{\mathcal{O}' \subseteq \mathcal{P} \wedge |\mathcal{O}'| \leq m} \sum_{p \in \mathcal{O}'} \Delta(p, \mathcal{C})$;
- 8 **return** \mathcal{C}, \mathcal{O} .

▷ **Claim 17.** If $|\mathcal{P}_i^f| > \frac{\varepsilon}{17} |\mathcal{P}_i|$, then $\Delta(\mathcal{P}_i, c_i) < (1 + \frac{\varepsilon}{5}) \Delta(\mathcal{P}_i) + \frac{3\varepsilon}{10k} \Delta(\mathbb{P})$.

Observe that

$$\{p \in \mathcal{P} : \Delta(p, \mathcal{C}_{i-1} \uplus \{c_i\}) > \frac{\varepsilon \Delta(\mathbb{P})}{8k |\mathcal{P}_i|}\} \subseteq \{p \in \mathcal{P} : \Delta(p, \mathcal{C}_{i-1}) > \frac{\varepsilon \Delta(\mathbb{P})}{8k |\mathcal{P}_i|}\} = \mathcal{P} \setminus \mathcal{B}_{i-1} \subseteq \mathcal{P}^\ddagger,$$

where the last step is due to Lemma 14. Combining this with Claim 17, we know that Lemma 16 is true. ◀

Lemma 12 and Lemma 16 suggest that for each $i \in \{2, \dots, k\}$, $\tau(i)$ holds if $\tau(i-1)$ is true. Combining this with the initial condition $\tau(1)$ being true, we establish the validity of $\tau(i)$ for each $i \in [k]$. The proof of Lemma 10 effortlessly follows from the statement of $\tau(k)$.

4 Applications

In this section we show the applications of Algorithm 1. We first address the Euclidean k -MEANSOUT problem, and then show how to extend our approach to constrained cases.

4.1 The Algorithm for Euclidean k -MeansOut

Our approach for solving Euclidean k -MEANSOUT is presented in Algorithm 2, which takes as inputs a constant $\varepsilon \in (0, 1)$ and an instance $\mathcal{I} = (\mathcal{P}, k, m)$ with $\mathcal{P} \subset \mathbb{R}^d$ and $|\mathcal{P}| = n$. The algorithm iteratively invokes Algorithm 1 to construct a collection of center sets and returns the one, along with the corresponding outlier set, that minimizes the cost for \mathcal{I} . By analyzing the performance of Algorithm 2, we complete the proof of Theorem 2.

Proof (of Theorem 2). Let $(\mathcal{C}^*, \mathcal{O}^*)$ be an optimal solution to \mathcal{I} , which opens a set $\mathcal{C}^* = \{c_1^*, \dots, c_k^*\}$ of k centers from \mathbb{R}^d and removes a set \mathcal{O}^* of m outliers from \mathcal{P} . For each $i \in [k]$, denote by \mathcal{P}_i^* the subset of the points in $\mathcal{P} \setminus \mathcal{O}^*$ whose closest center in \mathcal{C}^* is c_i^* . Define $\Delta(\mathbb{P}) = \sum_{i=1}^k \Delta(\mathcal{P}_i^*, c_i^*) = \sum_{p \in \mathcal{P} \setminus \mathcal{O}^*} \Delta(p, \mathcal{C}^*)$ as the cost of $(\mathcal{C}^*, \mathcal{O}^*)$. Moreover, let \mathbb{C} be the collection of center sets constructed by Algorithm 2, and let $(\mathcal{C}, \mathcal{O})$ be the solution to \mathcal{I} returned by the algorithm. Observe that the statement in Lemma 10 holds with probability no less than 15^{-k} . Given that Algorithm 2 invokes **Sampling** 15^k times to construct \mathbb{C} , the probability of the statement in Lemma 10 being true in at least one of these invocations can be lower-bounded by $1 - (1 - 15^{-k})^{15^k} > 1 - e^{-1}$. Consequently, inequality

$$\sum_{p \in \mathcal{P} \setminus \mathcal{O}} \Delta(p, \mathcal{C}) \leq \sum_{i=1}^k \min_{c \in \mathbb{C}} \Delta(\mathcal{P}_i^*, c) \leq (1 + \varepsilon) \Delta(\mathbb{P}) \quad (8)$$

holds with probability at least $1 - e^{-1}$, where the first step is due to the operation in step 9 of Algorithm 2, and the second step follows from Lemma 10. Inequality (8) implies that the approximation ratio of Algorithm 2 is $1 + \varepsilon$.

It remains to analyze the running time of Algorithm 2. Lemma 9 implies that invoking **Sampling** 15^k times takes $nd((k+m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ time and adds $n((k+m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ center sets to \mathbb{C} . For each center set from \mathbb{C} , the algorithm takes $O(ndk)$ time to compute the corresponding cost for \mathcal{I} . Consequently, we know that Algorithm 2 runs in $n^2d((k+m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ time. This completes the proof of Theorem 2. \blacktriangleleft

4.2 A Coreset-Based Accelerated Algorithm

In a recent study, Huang et al. [29] showed that a subset of $\text{poly}(m, k, \varepsilon^{-1})$ weighted points, referred to as a coreset, can be identified in linear time for the given instance of Euclidean k -MEANSOUT, such that the outlier-removal clustering costs, induced by any set of k centers, are similar between the coreset and the entire point set. Such a method for coreset construction can serve as a means to accelerate our algorithm. Specifically, when selecting a well-performing center set from the collection \mathbb{C} , we can compare the clustering costs induced by the center sets on the coreset rather than the entire point set. This reduces the running time of the algorithm in Theorem 2 to $nd((k+m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$, with an arbitrarily small loss in the approximation ratio.

► **Lemma 18** ([29]). *Given a constant $\varepsilon \in (0, 1)$, an instance $\mathcal{I} = (\mathcal{P}, k, m)$ of Euclidean k -MEANSOUT satisfying $\mathcal{P} \subset \mathbb{R}^d$ and $|\mathcal{P}| = n$, and an algorithm that has the guarantee of yielding a solution $(\mathcal{C}, \mathcal{O})$ satisfying $\sum_{p \in \mathcal{P} \setminus \mathcal{O}} \Delta(p, \mathcal{C}) \leq \alpha \cdot \text{opt}$, $|\mathcal{C}| \leq \beta k$, and $|\mathcal{O}| \leq \gamma m$ for three real numbers $\alpha, \beta, \gamma \geq 1$ in $T(n, d, k, m)$ time (opt is the cost of an optimal solution to \mathcal{I}), a weighted subset $\mathcal{S} \subseteq \mathcal{P}$ satisfying $|\mathcal{S}| \leq \gamma m + \beta(k\varepsilon^{-1})^{O(1)}$ with weight function $w : \mathcal{S} \rightarrow [0, +\infty)$ can be constructed in $T(n, d, k, m) + O(ndk)$ time, such that*

$$\min_{\mathcal{O}' \subseteq \mathcal{S} \wedge \sum_{p \in \mathcal{O}'} w(p) \leq m} \sum_{p \in \mathcal{S} \setminus \mathcal{O}'} w(p) \Delta(p, \mathcal{C}') \in (1 \pm \alpha\varepsilon) \min_{\mathcal{O}' \subseteq \mathcal{P} \wedge |\mathcal{O}'| \leq m} \sum_{p \in \mathcal{P} \setminus \mathcal{O}'} \Delta(p, \mathcal{C}')$$

for each $\mathcal{C}' \subset \mathbb{R}^d$ with $|\mathcal{C}'| = k$.

To leverage Lemma 18, we give a straightforward and fast bi-criteria approximation algorithm for Euclidean k -MEANSOUT, based on the D^2 -sampling method given by Arthur and Vassilvitskii [2].

► **Lemma 19.** *Given an instance $\mathcal{I} = (\mathcal{P}, k, m)$ of Euclidean k -MEANSOUT with $\mathcal{P} \subset \mathbb{R}^d$ and $|\mathcal{P}| = n$, a center set $\mathcal{C} \subset \mathbb{R}^d$ satisfying $|\mathcal{C}| = O(k+m)$ and $\sum_{p \in \mathcal{P}} \Delta(p, \mathcal{C}) \leq O(\text{opt})$ can be identified in $O(nd(k+m))$ time, where opt is the cost of an optimal solution to \mathcal{I} .*

Taking as an input the algorithm given in Lemma 19, Lemma 18 yields a coreset of size $((k+m)\varepsilon^{-1})^{O(1)}$ in $O(nd(k+m))$ time. Leveraging this coreset allows us to reduce the running time of our algorithm for Euclidean k -MEANSOUT to be linear in n , as described in the following theorem.

► **Theorem 20.** *Given a constant $\varepsilon \in (0, 1)$ and an instance (\mathcal{P}, k, m) of Euclidean k -MEANSOUT satisfying $\mathcal{P} \subset \mathbb{R}^d$ and $|\mathcal{P}| = n$, there is a $(1 + O(\varepsilon))$ -approximation algorithm running in $nd((k + m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ time.*

Proof. Let $\mathcal{S} \subseteq \mathcal{P}$ be the coreset constructed by Lemma 18 when taking as inputs constant ε , instance (\mathcal{P}, k, m) , and the algorithm in Lemma 19, and let $w : \mathcal{S} \rightarrow [0, +\infty)$ be the corresponding weight function. We have $|\mathcal{S}| \leq ((k + m)\varepsilon^{-1})^{O(1)}$ due to Lemma 18 and Lemma 19. Our accelerated algorithm for Euclidean k -MEANSOUT is identical to Algorithm 2, differing only in steps 6 and 8, where we calculate the outlier-removal clustering costs induced by the candidate center sets and select the one with minimum cost. We now define

$$\mathbf{wcost}(\mathcal{C}') = \min_{\mathcal{O}' \subseteq \mathcal{S} \wedge \sum_{p \in \mathcal{O}'} w(p) \leq m} \sum_{p \in \mathcal{S} \setminus \mathcal{O}'} w(p) \Delta(p, \mathcal{C}')$$

for each $\mathcal{C}' \in \mathbb{C}$ in step 6 of the algorithm, and select the center set $\mathcal{C} \in \mathbb{C}$ with minimum value of $\mathbf{wcost}(\mathcal{C})$ in step 8.

Observe that calculating $\mathbf{wcost}(\mathcal{C}')$ for each $\mathcal{C}' \in \mathbb{C}$ takes $O(|\mathcal{S}||\mathcal{C}'|d) \leq d((k + m)\varepsilon^{-1})^{O(1)}$ time. Combining this with the fact that we take $nd((k + m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ time to construct a collection \mathbb{C} of $n((k + m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ candidate center sets (as argued in the proof of Theorem 2), we know that the accelerated algorithm runs in $nd((k + m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ time.

The analysis of the approximation ratio of our accelerated algorithm follows that of Algorithm 2 (given in the proof of Theorem 2). Denote by $(\tilde{\mathcal{C}}, \tilde{\mathcal{O}})$ the solution to \mathcal{I} constructed by the accelerated algorithm, and let $(\mathcal{C}, \mathcal{O})$ be a solution to \mathcal{I} satisfying inequality (8). We have

$$\sum_{p \in \mathcal{P} \setminus \tilde{\mathcal{O}}} \Delta(p, \tilde{\mathcal{C}}) \leq \frac{\mathbf{wcost}(\tilde{\mathcal{C}})}{1 - O(\varepsilon)} \leq \frac{\mathbf{wcost}(\mathcal{C})}{1 - O(\varepsilon)} \leq \frac{1 + O(\varepsilon)}{1 - O(\varepsilon)} \sum_{p \in \mathcal{P} \setminus \mathcal{O}} \Delta(p, \mathcal{C}), \quad (9)$$

where the first and third steps follow from the approximation guarantee of the coreset given by Lemma 18 and Lemma 19, and the second step is due to the fact that the center set $\tilde{\mathcal{C}}$ selected by our accelerated algorithm satisfies $\mathbf{wcost}(\tilde{\mathcal{C}}) = \min_{\mathcal{C}' \in \mathbb{C}} \mathbf{wcost}(\mathcal{C}')$. Combining inequality (9) with inequality (8), we know that the approximation ratio of the accelerated algorithm is $1 + O(\varepsilon)$. ◀

5 Extensions to Constrained Cases

Constrained k -MEANSOUT problems generalize k -MEANSOUT by introducing additional constraints on the feasibilities of solutions. For example, in the capacitated generalization, there is an upper bound on the size of each cluster. Similarly, in the lower-bounded generalization, each cluster must contain at least a specified number of points. There are known frameworks for reducing constrained k -MEANSOUT problems to their outlier-free counterparts with small losses in the approximation ratios [32, 13]. Combined with the approximation schemes applicable in outlier-free cases, such as the ones given in [8, 16, 4], these frameworks enable the development of $(1 + \varepsilon)$ -approximation algorithms for constrained k -MEANSOUT problems. However, similar to the unconstrained case, the reductions given in [32, 13] require time exponentially dependent on m .

An apparent distinction between constrained k -MEANSOUT problems and the unconstrained counterpart lies in the locality property described in Section 1.1: In the former, clusters in an optimal solution can be quite different from the ones with minimum clustering cost, as additional constraints are imposed on their feasibilities. Fortunately, Lemma 10

implies that a near-optimal set of centers corresponding to any outlier-removal k -clustering result can be found by Algorithm 1, including those imposed with additional constraints. Given k good-enough centers, it is sufficient to remove the outliers and assign each point to a center under the additional constraints. Indeed, combining Algorithm 1 with problem-specific methods for identifying outliers and assigning points, we obtain the first PTASs for constrained k -MEANSOUT problems in Euclidean spaces for constant k and super-constant m , including capacitated and fair k -MEANSOUT.

Our algorithms for constrained k -MEANSOUT problems closely resemble Algorithm 2, with the only difference being in the construction of the solution based on \mathbb{C} (as shown in steps 5-10). Given a set $\mathcal{P} \subset \mathbb{R}^d$ with $|\mathcal{P}| = n$, a real number $\varepsilon \in (0, 1]$, and two positive integers k and m , let \mathbb{C} be the collection of center sets constructed by Algorithm 2. By Lemma 10 and the fact that \mathbb{C} is constructed by invoking **Sampling** 15^k times, we know that the following event happens with probability no less than $1 - (1 - 15^{-k})^{15^k} > 1 - e^{-1}$: For any k disjoint subsets $\mathcal{P}_1, \dots, \mathcal{P}_k$ of \mathcal{P} with $\sum_{i=1}^k |\mathcal{P}_i| = n - m$, there is a center set $\mathcal{C} \in \mathbb{C}$ satisfying

$$\sum_{i=1}^k \min_{c \in \mathcal{C}} \Delta(\mathcal{P}_i, c) \leq (1 + \varepsilon) \sum_{i=1}^k \Delta(\mathcal{P}_i). \quad (10)$$

This implies a good chance of finding a near-optimal set of centers corresponding to any outlier-removal k -clustering result in \mathbb{C} . The next step involves developing problem-specific methods to remove m outliers and partition the remaining points into k clusters based on the given center set, with the objective of minimizing the clustering cost (defined as the sum of the squared distances from the points to the corresponding centers) while satisfying the specified additional constraints. If we can remove the outliers and partition the points in an optimal way, then the guarantee of the center set given in inequality (10) suggests the achievement of a $(1 + \varepsilon)$ -approximation solution.

5.1 The Algorithm for Capacitated k -MeansOut

Capacitated clustering is one of the most extensively studied generalizations of the standard clustering formulation, which imposes an upper-bound constraint on the size of each cluster. An Euclidean instance of capacitated k -MEANSOUT is specified by a set $\mathcal{P} \subset \mathbb{R}^d$ of n points, two positive integers k and m , and a *capacity* $u \geq 1$. A feasible solution to the instance selects a set $\mathcal{C} \subset \mathbb{R}^d$ of centers and a set $\mathcal{O} \subseteq \mathcal{P}$ of outliers, and assigns each point $p \in \mathcal{P} \setminus \mathcal{O}$ to a center $\varphi(p) \in \mathcal{C}$, such that $|\mathcal{C}| \leq k$, $|\mathcal{O}| \leq m$, and $|\varphi^{-1}(c)| \leq u$ for each $c \in \mathcal{C}$. The cost of such a solution is $\sum_{p \in \mathcal{P} \setminus \mathcal{O}} \|p - \varphi(p)\|^2$. The goal of the problem is to find a feasible solution with minimum cost.

Motivated by the method for solving outlier-free constrained problems given in [16], we reduce the task of identifying outliers and assigning points for capacitated k -MEANSOUT to the well-known *minimum-cost circulation* problem [40], which can be defined as follows.

► **Definition 21** (minimum-cost circulation). *An instance of the minimum-cost circulation problem is specified by a directed graph $G(\mathcal{V}, \mathcal{A})$ with a set \mathcal{V} of vertices and a set \mathcal{A} of arcs, where each $(v, w) \in \mathcal{A}$ has a cost $\Delta(v, w) \geq 0$, a capacity $u(v, w) \geq 0$, and a demand $l(v, w) \in [0, u(v, w)]$. A feasible solution to the instance associates each $(v, w) \in \mathcal{A}$ with a flow $f(v, w) \in [l(v, w), u(v, w)]$ such that $\sum_{w: (v, w) \in \mathcal{A}} f(v, w) = \sum_{w: (w, v) \in \mathcal{A}} f(w, v)$ for each $v \in \mathcal{V}$. The cost of this solution is $\sum_{(v, w) \in \mathcal{A}} \Delta(v, w) f(v, w)$. The problem aims to find a feasible solution with minimum cost.*

84:14 Faster Approximation Schemes for (Constrained) k -Means with Outliers

Let $\mathcal{I} = (\mathcal{P}, k, m, u)$ be an Euclidean instance of capacitated k -MEANSOUT. As previously discussed, Algorithm 2 can construct a collection \mathbb{C} of center sets, including a near-optimal one for \mathcal{I} , with high probability. For each $\mathcal{C} \in \mathbb{C}$, we construct an instance of minimum-cost circulation as follows.

- The vertex set \mathcal{V} consists of the points in \mathcal{P} , the centers in \mathcal{C} , and three additional vertices v_1, v_2 , and v_3 .
- There is an arc $(v_3, v_1) \in \mathcal{A}$ with $\Delta(v_3, v_1) = 0$ and $u(v_3, v_1) = l(v_3, v_1) = |\mathcal{P}|$.
- For each $p \in \mathcal{P}$ and $c \in \mathcal{C}$, there is an arc $(p, c) \in \mathcal{A}$ with $\Delta(p, c) = \|p - c\|^2$, $u(p, c) = 1$, and $l(p, c) = 0$. Here, $f(p, c) = 1$ signifies the assignment of point p to center c (i.e., $\varphi(c) = p$).
- For each $p \in \mathcal{P}$, there is an arc $(p, v_2) \in \mathcal{A}$ with $\Delta(p, v_2) = 0$, $u(p, v_2) = 1$, and $l(p, v_2) = 0$. A point p with $f(p, v_2) = 1$ is identified as an outlier. To ensure that the outliers are no more than m (more formally, $\sum_{p \in \mathcal{P}} f(p, v_2) \leq m$), we add to \mathcal{A} an arc (v_2, v_3) with $\Delta(v_2, v_3) = 0$, $u(v_2, v_3) = m$, and $l(v_2, v_3) = 0$.
- To ensure that each $p \in \mathcal{P}$ is assigned to a center in \mathcal{C} or identified as an outlier (i.e., $\sum_{c \in \mathcal{C} \cup \{v_2\}} f(p, c) = 1$), we add to \mathcal{A} an arc (v_1, p) with $\Delta(v_1, p) = 0$ and $u(v_1, p) = l(v_1, p) = 1$.
- To satisfy the capacity constraint imposed on each $c \in \mathcal{C}$ (i.e., $\sum_{p \in \mathcal{P}} f(p, c) \leq u$), we add to \mathcal{A} an arc (c, v_3) with $\Delta(c, v_3) = 0$, $u(c, v_3) = \lfloor u \rfloor$, and $l(c, v_3) = 0$.

Given that all the capacities and demands in the instance of minimum-cost circulation described above are integers, its optimal integral solutions can be found in $(nk)^{O(1)}$ time [40]. It can be seen that these solutions correspond to optimal ways of identifying outliers and assigning points for the given center set.

Let $(\mathcal{C}^*, \mathcal{O}^*, \varphi^*)$ be an optimal solution to \mathcal{I} , where $\mathcal{C}^* = \{c_1^*, \dots, c_k^*\}$. For each $i \in [k]$, define $\mathcal{P}_i^* = \{p \in \mathcal{P} \setminus \mathcal{O}^* : \varphi^*(p) = c_i^*\}$. Inequality (10) implies that there is a center set $\mathcal{C} \in \mathbb{C}$ satisfying

$$\sum_{i=1}^k \min_{c \in \mathcal{C}} \Delta(\mathcal{P}_i^*, c) \leq (1 + \varepsilon) \sum_{i=1}^k \Delta(\mathcal{P}_i^*) = (1 + \varepsilon) \sum_{p \in \mathcal{P} \setminus \mathcal{O}^*} \|p - \varphi^*(p)\|^2 \quad (11)$$

with constant probability. Based on an optimal integral solution to the instance of minimum-cost circulation corresponding to such a center set \mathcal{C} , we can construct a solution $(\mathcal{C}, \mathcal{O}, \varphi)$ to \mathcal{I} satisfying

$$\sum_{p \in \mathcal{P} \setminus \mathcal{O}} \|p - \varphi(p)\|^2 \leq \sum_{i=1}^k \min_{c \in \mathcal{C}} \Delta(\mathcal{P}_i^*, c). \quad (12)$$

Inequalities (11) and (12) imply that a $(1 + \varepsilon)$ -approximation solution to \mathcal{I} has been constructed. Recall that \mathbb{C} is of size $n((k + m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ and can be constructed in $nd((k + m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ time (due to Lemma 9). Combining this with the fact that the instance of minimum-cost circulation corresponding to each center set in \mathbb{C} can be solved in $(nk)^{O(1)}$ time, we know that constructing the $(1 + \varepsilon)$ -approximation solution takes $n^{O(1)}d((k + m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ time.

► **Theorem 22.** *Given a constant $\varepsilon \in (0, 1)$ and an Euclidean instance (\mathcal{P}, k, m, u) of capacitated k -MEANSOUT satisfying $\mathcal{P} \subset \mathbb{R}^d$ and $|\mathcal{P}| = n$, there is a $(1 + \varepsilon)$ -approximation algorithm running in $n^{O(1)}d((k + m)\varepsilon^{-1})^{O(k\varepsilon^{-1})}$ time.*

5.2 The Algorithm for Fair k -MeansOut

Motivated by applications related to fair data representation [5], fair clustering problems, which impose constraints on the proportions of each type of points within the clusters, have garnered significant attention. We consider the fair k -MEANSOUT problem defined in [13]. An instance of the problem is specified by a collection $\mathbb{P} = \{\mathcal{P}^1, \dots, \mathcal{P}^\ell\}$ of ℓ disjoint sets of points in \mathbb{R}^d , a positive integer k , two fairness vectors $\vec{\alpha}, \vec{\beta} \in [0, 1]^\ell$, and an outlier vector $\vec{m} \in \mathbb{N}^\ell$. A feasible solution to this instance selects a set $\mathcal{C} \subset \mathbb{R}^d$ of no more than k centers and a set $\mathcal{O} \subseteq \bigcup_{i=1}^\ell \mathcal{P}^i$ of outliers, and assigns each point $p \in \bigcup_{i=1}^\ell \mathcal{P}^i \setminus \mathcal{O}$ to a center $\varphi(p) \in \mathcal{C}$, such that $|\mathcal{P}^i \cap \varphi^{-1}(c)| |\varphi^{-1}(c)|^{-1} \in [\alpha_i, \beta_i]$ and $|\mathcal{P}^i \cap \mathcal{O}| \leq m_i$ for each $i \in [\ell]$ and $c \in \mathcal{C}$. The cost of the solution is $\sum_{i=1}^\ell \sum_{p \in \mathcal{P}^i} \|p - \varphi(p)\|^2$.

Similar to our previous strategy, we address the fair k -MEANSOUT problem based on a collection of center sets constructed by repeatedly invoking Algorithm 1. Unfortunately, identifying outliers and assigning points for a given set of centers in an optimal way within polynomial time is no longer feasible. Specifically, a reduction from the $3D$ -matching problem suggests that this task is NP-hard [5]. Using the mixed-integer linear programming-based algorithm given in [13], this task can be completed in an exponential time.

► **Lemma 23** ([13]). *Given a set $\mathcal{C} \subset \mathbb{R}^d$ of no more than k centers and an instance $\mathcal{I} = (\mathbb{P}, k, \vec{\alpha}, \vec{\beta}, \vec{m})$ of fair k -MEANSOUT satisfying $|\mathbb{P}| = \ell$, $\sum_{\mathcal{P} \in \mathbb{P}} |\mathcal{P}| = n$, $\sum_{i=1}^\ell m_i = m$, and $\mathcal{P} \subset \mathbb{R}^d$ for each $\mathcal{P} \in \mathbb{P}$, a feasible solution to \mathcal{I} with minimum cost among the ones taking \mathcal{C} as the center set can be constructed in $(k\ell)^{O(k\ell)} n^{O(1)} dL$ time, where L is the bit-size of \mathcal{I} .*

Using our sampling-based algorithm for constructing center sets (m is replaced with $\sum_{i=1}^\ell m_i$) and the algorithm for constructing solutions given in Lemma 23, we obtain the following approximation scheme for fair k -MEANSOUT.

► **Theorem 24.** *Given a constant $\varepsilon \in (0, 1)$ and an instance $\mathcal{I} = (\mathbb{P}, k, \vec{\alpha}, \vec{\beta}, \vec{m})$ of fair k -MEANSOUT satisfying $|\mathbb{P}| = \ell$, $\sum_{\mathcal{P} \in \mathbb{P}} |\mathcal{P}| = n$, $\sum_{i=1}^\ell m_i = m$, and $\mathcal{P} \subset \mathbb{R}^d$ for each $\mathcal{P} \in \mathbb{P}$, there is a $(1 + \varepsilon)$ -approximation algorithm running in $n^{O(1)} d((k + m)\varepsilon^{-1})^{O(k\varepsilon^{-1})} (k\ell)^{O(k\ell)} L$ time, where L is the bit-size of \mathcal{I} .*

6 Conclusions

In this paper, we present $(1 + \varepsilon)$ -approximation algorithms with running times exponential in k for Euclidean k -MEANSOUT and constrained generalizations of the problem, including capacitated and fair k -MEANSOUT. For each considered problem, our proposed algorithm stands for the first PTAS for constant k .

Considering the APX-hardness of Euclidean k -MEANSOUT [3], it is unlikely to design a $(1 + \varepsilon)$ -approximation algorithm without exponential dependence on k in high dimensions. Nonetheless, exploring ways to improve the running time of our algorithm for Euclidean k -MEANSOUT remains an interesting direction for future research. Especially, one can see whether it is possible to develop a $(1 + \varepsilon)$ -approximation algorithm running in $(ndm)^{O(1)} f(k, \varepsilon)$ time for some positive function f .

References

- 1 Akanksha Agrawal, Tanmay Inamdar, Saket Saurabh, and Jie Xue. Clustering what matters: Optimal approximation for clustering with outliers. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pages 6666–6674, 2023.

- 2 David Arthur and Sergei Vassilvitskii. k -means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, 2007.
- 3 Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The hardness of approximation of Euclidean k -means. In *Proceedings of the 31st International Symposium on Computational Geometry*, volume 34, pages 754–767, 2015.
- 4 Sayan Bandyapadhyay, Fedor V. Fomin, and Kirill Simonov. On coresets for fair clustering in metric and Euclidean spaces and their applications. In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming*, volume 198, pages 23:1–23:15, 2021.
- 5 Suman Kalyan Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems*, pages 4955–4966, 2019.
- 6 Aditya Bhaskara, Sharvaree Vadgama, and Hong Xu. Greedy sampling for approximate clustering in the presence of outliers. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems*, pages 11146–11155, 2019.
- 7 Anup Bhattacharya, Dishant Goyal, Ragesh Jaiswal, and Amit Kumar. On sampling based algorithms for k -means. In *Proceedings of the 40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 182, pages 13:1–13:17, 2020.
- 8 Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Faster algorithms for the constrained k -means problem. *Theory Comput. Syst.*, 62(1):93–115, 2018.
- 9 Sanjay Chawla and Aristides Gionis. k -means--: A unified approach to clustering and outlier detection. In *Proceedings of the 13th SIAM International Conference on Data Mining*, pages 189–197, 2013.
- 10 Jiecao Chen, Erfan Sadeqi Azer, and Qin Zhang. A practical algorithm for distributed clustering and outlier detection. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pages 2253–2262, 2018.
- 11 Ke Chen. On coresets for k -median and k -means clustering in metric and Euclidean spaces and their applications. *SIAM J. Comput.*, 39(3):923–947, 2009.
- 12 Vincent Cohen-Addad, Andreas Emil Feldmann, and David Saulpic. Near-linear time approximation schemes for clustering in doubling metrics. *J. ACM*, 68(6):44:1–44:34, 2021.
- 13 Rajni Dabas, Neelima Gupta, and Tanmay Inamdar. FPT approximations for capacitated/fair clustering with outliers. *CoRR*, abs/2305.01471, 2023.
- 14 Wenceslas Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani. Approximation schemes for clustering problems. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 50–58. ACM, 2003.
- 15 Amit Deshpande, Praneeth Kacham, and Rameshwar Pratap. Robust k -means++. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence*, volume 124, pages 799–808, 2020.
- 16 Hu Ding and Jinhui Xu. A unified framework for clustering constrained data without locality property. *Algorithmica*, 82(4):808–852, 2020.
- 17 Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the 43rd ACM Symposium on Theory of Computing*, pages 569–578, 2011.
- 18 Dan Feldman, Morteza Monemizadeh, and Christian Sohler. A PTAS for k -means clustering based on weak coresets. In *Proceedings of the 23rd ACM Symposium on Computational Geometry*, pages 11–18, 2007.
- 19 Dan Feldman and Leonard J. Schulman. Data reduction for weighted and outlier-resistant clustering. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1343–1354, 2012.
- 20 Zachary Frigstad, Kamyar Khodamoradi, Mohsen Rezapour, and Mohammad R. Salavatipour. Approximation schemes for clustering with outliers. *ACM Trans. Algorithms*, 15(2):26:1–26:26, 2019.

- 21 Luis Ángel García-Escudero and Alfonso Gordaliza. Robustness properties of k -means and trimmed k -means. *J. Am. Stat. Assoc.*, 94(447):956–969, 1999.
- 22 Alexandros Georgogiannis. Robust k -means: A theoretical revisit. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems*, pages 2883–2891, 2016.
- 23 Christoph Grunau and Václav Rozhon. Adapting k -means algorithms for outliers. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 7845–7886, 2022.
- 24 Sudipto Guha, Yi Li, and Qin Zhang. Distributed partial clustering. *ACM Trans. Parallel Comput.*, 6(3):11:1–11:20, 2019.
- 25 Shalmoli Gupta, Ravi Kumar, Kefu Lu, Benjamin Moseley, and Sergei Vassilvitskii. Local search methods for k -means with outliers. *Proc. VLDB Endow.*, 10(7):757–768, 2017.
- 26 Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *J. Am. Stat. Assoc.*, 58:13–30, 1963.
- 27 Junyu Huang, Qilong Feng, Ziyun Huang, Jinhui Xu, and Jianxin Wang. Fast algorithms for distributed k -clustering with outliers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 13845–13868, 2023.
- 28 Lingxiao Huang, Shaofeng H.-C. Jiang, Jian Li, and Xuan Wu. ϵ -coresets for clustering (with outliers) in doubling metrics. In *Proceedings of the 59th IEEE Annual Symposium on Foundations of Computer Science*, pages 814–825, 2018.
- 29 Lingxiao Huang, Shaofeng H.-C. Jiang, Jianing Lou, and Xuan Wu. Near-optimal coresets for robust clustering. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.
- 30 Sungjin Im, Mahshid Montazer Qaem, Benjamin Moseley, Xiaorui Sun, and Rudy Zhou. Fast noise removal for k -means clustering. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108, pages 456–466, 2020.
- 31 Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted Voronoi diagrams and randomization to variance-based k -clustering (extended abstract). In *Proceedings of the 10th Annual Symposium on Computational Geometry*, pages 332–339, 1994.
- 32 Ragesh Jaiswal and Amit Kumar. Clustering what matters in constrained settings: Improved outlier to outlier-free reductions. In *Proceedings of the 34th International Symposium on Algorithms and Computation*, volume 283, pages 41:1–41:16, 2023.
- 33 Ragesh Jaiswal, Amit Kumar, and Sandeep Sen. A simple D^2 -sampling based PTAS for k -means and other clustering problems. *Algorithmica*, 70(1):22–46, 2014.
- 34 Ragesh Jaiswal, Mehul Kumar, and Pulkit Yadav. Improved analysis of D^2 -sampling based PTAS for k -means and other clustering problems. *Inf. Process. Lett.*, 115(2):100–103, 2015.
- 35 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.*, 28(2-3):89–112, 2004.
- 36 Ravishankar Krishnaswamy, Shi Li, and Sai Sandeep. Constant approximation for k -median and k -means with outliers via iterative rounding. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 646–659, 2018.
- 37 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.
- 38 Euiwoong Lee, Melanie Schmidt, and John Wright. Improved and simplified inapproximability for k -means. *Inf. Process. Lett.*, 120:40–43, 2017.
- 39 Shi Li and Xiangyu Guo. Distributed k -clustering for data with heavy noise. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pages 7849–7857, 2018.
- 40 Alexander Schrijver. *Combinatorial optimization: Polyhedra and efficiency*. Springer, Berlin, 2003.
- 41 Zhen Zhang, Qilong Feng, Junyu Huang, Yutian Guo, Jinhui Xu, and Jianxin Wang. A local search algorithm for k -means with outliers. *Neurocomputing*, 450:230–241, 2021.