Approximation Algorithms for Cost-Balanced Clustering

Amit Deshpande
Microsoft Research
amitdesh@microsoft.com

Anand Louis
IISc
anandl@iisc.ac.in

Deval Patel
IISc
devalpatel@iisc.ac.in

Apoorv Vikram Singh
IISc
apoorvsingh@iisc.ac.in

Abstract

Clustering points in the Euclidean space is a fundamental problem in the theory of algorithms and in unsupervised learning. Various clustering objectives to quantify the quality of clustering have been proposed and studied; the k-means and k-median clustering objective are the most popular ones. In some cases, the k-means or the k-median objective may result in a few clusters of very large cost and many clusters of extremely small cost. Even when the optimal clusters are balanced in size, some of them may have a huge variance. This is undesirable for quantization or when we have a budget constraint on the cost of each cluster. Motivated by this, we study the cost-balanced k-means and the cost-balanced k-median problem. For a k-clustering $O_1, \ldots, O_k$ of a given set of n points $X \subset \mathbb{R}^d$, we define its cost-balanced k-means cost as

$$\mathcal{K}(O_1, \ldots, O_k) \stackrel{\text{def}}{=} \max_{l \in [k]} \sum_{x \in O_l} \|x - \mu_l\|^2 \qquad \text{where } \mu_l = \frac{1}{|O_l|} \sum_{x \in O_l} x\,.$$

In other words, we want to minimize the cost of the heaviest cluster or balance the cost of each cluster. For any $\varepsilon > 0$, we give a randomized algorithm with running time $O\left(2^{\mathsf{poly}(k/\varepsilon)} nd\right)$ that gives a $(1+\varepsilon)$-approximation to the optimal cost-balanced k-means and the similarly defined optimal cost-balanced k-median clustering, using k clusters, with a constant probability. We define a more general version of the k-median clustering and the cost-balanced k-median clustering, and we name them ℓ_p cost k-clustering and ℓ_p cost-balanced k-clustering, respectively. Given a black-box algorithm which gives a constant factor approximation to the ℓ_p cost k-clustering, we show a procedure that runs in time $\mathsf{poly}(n, k, p)$ which gives a bi-criteria $O\left(1/\varepsilon^{1/p}\right)$-approximation to the optimal ℓ_p cost-balanced k-clustering, using $(1+\varepsilon)k$ clusters.

1 Introduction

Clustering points in the Euclidean space is a fundamental problem in the theory of algorithms and in unsupervised learning. Given a set of points, the goal is to group "similar" points together. Typically, a number k is also provided as an input, and the problem asks to find k clusters in the set of points.

To quantify the quality of clustering, various clustering objectives have been studied, such as k-means, k-median, k-center, etc. Out of these, k-means and k-median are one of the most popular clustering objectives. Given a set of points X in $\mathbb{R}^d$, and a clustering $\{O_1, \ldots, O_k\}$ of X, the k-means cost of the clustering is defined as

$$k\text{-means}\,(O_1, \ldots, O_k) \stackrel{\text{def}}{=} \sum_{l=1}^{k} \sum_{x \in O_l} \|x - \mu_l\|^2 \qquad \text{where } \mu_l \stackrel{\text{def}}{=} \frac{1}{|O_l|} \sum_{x \in O_l} x\,, \tag{1}$$

and the k-median cost of the clustering is defined as

$$k\text{-median}\,(O_1, \ldots, O_k) \stackrel{\text{def}}{=} \sum_{l=1}^{k} \sum_{x \in O_l} \|x - c_l\| \qquad \text{where } c_l \stackrel{\text{def}}{=} \mathsf{argmin}_{c \in \mathbb{R}^d} \sum_{x \in O_l} \|x - c_l\|\,. \tag{2}$$

Computing a clustering having the least k-means cost and the least k-median cost is an NP-hard problem. There have been many works studying approximation algorithms for this problem, see Section 1.2 for a brief survey.

Another popular clustering objective is the k-center clustering objective. For a set of points X in $\mathbb{R}^d$, the k-center problem asks to compute a set C of k points in $\mathbb{R}^d$, which minimize $\max_{x \in X} \min_{c \in C} \|x - c\|$. This problem can be equivalently stated as a clustering problem as follows: compute a k-clustering $O_1, \ldots, O_k$ of X and "cluster centers" $c_1, \ldots, c_k \in \mathbb{R}^d$ so as to minimize

$$\max_{l \in [k]} \max_{x \in O_l} \|x - c_l\|\,.$$

In some cases, the k-means and the k-median objective may result in a few clusters of very large cost and many clusters of extremely small cost. Even when the optimal clusters are balanced in size, some of them may have huge variance. This is undesirable for quantization, or when we have a budget constraint on the cost of each cluster. The k-center clustering objective may result in a some clusters being unbalanced (i.e. sizes of some clusters being significantly larger than the sizes of other clusters). Moreover, the k-center cost of a clustering, which can be viewed as a proxy for the *coverage cost* of a cluster, is very sensitive to outliers. Motivated by this, we study the *cost-balanced k-means* and the *cost-balanced k-median* clustering objective which is defined as follows.

Definition 1.1. Given a set X of n points in $\mathbb{R}^n$, and a clustering $O_1, \ldots, O_k$ of X, the cost-balanced k-means cost of the clustering is defined as

$$\mathcal{K}\,(O_1, \ldots, O_k) \stackrel{\text{def}}{=} \max_{l \in [k]} \sum_{x \in O_l} \|x - \mu_l\|^2 \qquad \text{where } \mu_l \stackrel{\text{def}}{=} \frac{1}{|O_l|} \sum_{x \in O_l} x\,,$$

and

Definition 1.2. Given a set X of n points in $\mathbb{R}^n$, and a clustering $O_1, \ldots, O_k$ of X, the cost-balanced k-median cost of the clustering is defined as

$$\mathcal{K}_{\text{med}}\,(O_1, \ldots, O_k) \stackrel{\text{def}}{=} \max_{l \in [k]} \sum_{x \in O_l} \|x - c_l\| \qquad \text{where } c_l \stackrel{\text{def}}{=} \mathsf{argmin}_{c \in \mathbb{R}^d} \sum_{x \in O_l} \|x - c_l\|\,.$$

The k-means cost of a clustering $O_1, \ldots, O_k$ can be viewed as the sum of 1-means cost of each cluster, whereas the cost-balanced k-means cost is defined as the largest 1-means cost among the k clusters. For a cluster, the sum of the squared distances of the points in the cluster to its center can be viewed as a proxy for the *coverage cost* of a cluster. This is more resilient to outliers, particularly in the case of balanced clusters. Minimizing the largest coverage cost among the k clusters gives the cost-balanced k-means problem. Therefore, the cost-balanced k-means clustering can be viewed as having the desirable properties of k-means and k-center. The similar motivation holds for the cost-balanced k-median objective.

We define a more general notion of k-means and cost-balanced k-means which generalizes to other well-known clustering objectives. Given a metric space (X, dist), where X represents the set of points and $\text{dist}(u, v)$ is the distance between $u, v \in X$.

Definition 1.3 (ℓ_p cost k-clustering cost)**.** For a k clustering $O_1, ..., O_k$ with corresponding centers $c_1, ..., c_k$, we define its ℓ_p cost k-clustering cost $\Delta_p((O_1, ..., O_k), (c_1, ..., c_k))$ as $\sum_{i=1}^{k} \sum_{u \in O_i} \text{dist}(u, c_i)^p$.

The optimal ℓ_p cost k-clustering cost is defined as

$$\min_{((O_1,...,O_k),(c_1,...,c_k))} \left(\sum_{i=1}^{k} \sum_{u \in O_i} \text{dist}(u, c_i)^p \right)^{\frac{1}{p}}.$$

Definition 1.4 (ℓ_p cost-balanced k-clustering cost)**.** For a k clustering $O_1, ..., O_k$ with corresponding centers $c_1, ..., c_k$, we define its ℓ_p cost-balanced k-clustering cost $\mathcal{K}_p((O_1, ..., O_k), (c_1, ..., c_k))$ as $\max_{i \in [k]} \sum_{u \in O_i} \text{dist}(u, c_i)^p$.

The optimal ℓ_p cost-balanced k-clustering cost is defined as

$$\min_{((O_1,...,O_k),(c_1,...,c_k))} \left(\max_{i \in [k]} \sum_{u \in O_i} \text{dist}(u, c_i)^p \right)^{\frac{1}{p}}.$$

Remark 1.5. For $p = 1$, ℓ_p cost-balanced k-clustering cost corresponds to cost-balanced k-median cost. For $p = 2$, ℓ_p cost-balanced k-clustering corresponds to cost-balanced k-means clustering and the ℓ_p cost-balanced k-clustering cost corresponds to *(cost-balanced k-means cost)*$^{\frac{1}{2}}$.

We study the problem of computing the k-clustering which has the least ℓ_p cost-balanced k-clustering cost. A cost-balanced-type objective is useful in facility-location, when each facility incurs the cost of serving its clients, and we would like to minimize the maximum service cost over all open facilities. In such applications, the optimal solution need not assign clients to their nearest facilities. cost-balanced versions of other clustering objectives have been studied before, see Section 1.2 for a brief survey.

1.1 Our Results

We give a $(1 + \varepsilon)$ approximation algorithm (Algorithm 1) for cost-balanced k-means problem, running in time $O\left(2^{\text{poly}(k/\varepsilon)} nd\right)$.

Theorem 1.6. *There exists a randomized polynomial time algorithm which, given a set X of n points in $\mathbb{R}^d$, an integer $k \in \mathbb{Z}$, and an error parameter $\varepsilon > 0$, runs in time $O\left(2^{\text{poly}(k/\varepsilon)} nd\right)$ and outputs, with constant probability, a k-partition $O_1, \ldots, O_k$ of X which satisfies $\mathcal{K}(O_1, \ldots, O_k) \leqslant (1 + \varepsilon)\,\mathsf{OPT}$, where OPT is the minimum value of $\mathcal{K}(O_1, \ldots, O_k)$ over all k-partitions $O_1, \ldots, O_k$ of X.*

We also give a $(1 + \varepsilon)$ approximation algorithm (Algorithm 4) for cost-balanced k-median problem, running in time $O\left(2^{\text{poly}(k/\varepsilon)} nd\right)$.

Theorem 1.7. *There exists a randomized polynomial time algorithm which, given a set X of n points in $\mathbb{R}^d$, an integer $k \in \mathbb{Z}$, and an error parameter $\varepsilon > 0$, runs in time $O\left(2^{\text{poly}(k/\varepsilon)} nd\right)$ and outputs, with constant probability, a k-partition $O_1, \ldots, O_k$ of X which satisfies $\mathcal{K}_{med}(O_1, \ldots, O_k) \leqslant (1 + \varepsilon)\,\mathsf{OPT}$, where OPT is the minimum value of $\mathcal{K}_{med}(O_1, \ldots, O_k)$ over all k-partitions $O_1, \ldots, O_k$ of X.*

We also give a polynomial time bi-criteria approximation algorithm (Algorithm 3) for the ℓ_p cost-balanced k-clustering cost minimization problem.

Theorem 1.8. *Given a set of n points in a metric space $(X, dist)$, an integer $k \in \mathbb{Z}$, a number $p \in \mathbb{R}_+$, and a polynomial running time algorithm $\mathcal{G}$ which outputs the clustering $\mathbb{O} = \{O_1, ..., O_k\}$ with centers $\mathbb{C} = \{c_1, ..., c_k\}$ such that it $\Delta_p(\mathbb{O}, \mathbb{C})$ is at most α times the optimal ℓ_p cost k-clustering cost, there exists a polynomial time algorithm that runs in time $O\left(\frac{n(\log k + p \log \alpha)}{\log(1+\xi)}\right)$, where $\xi > 0$ is an error parameter, and outputs a $(1 + \varepsilon)k$-partition, for $\varepsilon > 0$, $O^{\mathcal{A}}_1, \ldots, O^{\mathcal{A}}_{(1+\varepsilon)k}$ with corresponding centers $c^{\mathcal{A}}_1, ..., c^{\mathcal{A}}_{(1+\varepsilon)k}$ of X which satisfies $\max_{l \in [(1+\varepsilon)k]} \sum_{x \in O^{\mathcal{A}}_l} dist\left(x, c^{\mathcal{A}}_l\right)^p \leqslant \left(\frac{2\alpha(1+\xi)}{\varepsilon}\right) \cdot \mathsf{OPT}$, where OPT is the optimal ℓ_p cost-balanced k-clustering cost.*

Remark 1.9. The centers in the Theorem 1.8 need not belong to X. The theorem holds for both discrete ℓ_p cost-balanced k-clustering, where centers belong to X, and Euclidean ℓ_p cost-balanced k-clustering, where centers belong to $\mathbb{R}^d$.

As a consequence of the above theorem, we get the following corollaries.

Corollary 1.10. *For parameters $\xi > 0$, $\varepsilon > 0$, there exists an algorithm that runs in time $\mathsf{poly}\left(n, p, k, \frac{1}{\log(1+\xi)}\right)$, which gives a $\left(O\left(\frac{C(1+\xi)^{1/p}}{\varepsilon^{1/p}}\right), (1+\varepsilon)\right)$-bi-criteria approximation to the discrete version of the ℓ_p cost-balanced k-clustering clustering, for $p \geqslant 1$, $C > 0$, by using the $O(C)$-factor approximation algorithm of [CS19] for the ℓ_p cost k-clustering problem, for some constant $C > 1$.*

Corollary 1.11. *For parameteres $\xi > 0$, $\varepsilon > 0$, there exists an algorithm that runs in time $\mathsf{poly}\left(n, k, \frac{1}{\log(1+\xi)}\right)$, which gives a $\left(\left(12.75\frac{(1+\xi)}{\varepsilon}\right), (1+\varepsilon)\right)$-bi-criteria approximation to the discrete version of the cost-balanced k-means clustering by using the 6.375-factor approximation algorithm of [ANFSW17] for the discrete k-means problem.*

Corollary 1.12. *For parameteres $\xi > 0$, $\varepsilon > 0$, there exists an algorithm that runs in time $\mathsf{poly}\left(n, k, \frac{1}{\log(1+\xi)}\right)$, which gives a $\left(\left(5.35\frac{(1+\xi)}{\varepsilon}\right), (1+\varepsilon)\right)$-bi-criteria approximation to the discrete version of the cost-balanced k-median clustering by using the 2.675-factor approximation algorithm of [BPR+17] for the discrete k-median problem.*

Complementing these results, we prove that even if we are provided a set of k centers, the problem of computing the optimal clustering for these centers is NP-hard.

Proposition 1.13. *Given a set of n points $X \subset \mathbb{R}^d$, an integer $k \in \mathbb{Z}$, and k points $c_1, \ldots, c_k \in \mathbb{R}^d$, it is NP-hard to compute a k-partition $O_1, \ldots, O_k$ which minimizes $\max_{l \in [k]} \sum_{x \in O_l} \|x - c_l\|^2$.*

Organization We given an overview of the proofs of our results in Section 1.3. We prove Theorem 1.6 in Section 2, Theorem 1.7 in Section C, and Theorem 1.8 in Section 3. We prove Proposition 1.13 and Section 4.

1.2 Related Work

Approx Algorithms for k-means and k-median Kanungo et al. [KMN+04] proposed a $(9+\varepsilon)$-approximation algorithm with running time $O\left(n^3\varepsilon^{-d}\right)$ for the discrete version of the k-means problem. Arthur and Vassilvitskii [AV07] showed an approximation ratio of $O(\log k)$, with running time $O(ndk)$, much superior to [KMN+04], for the Euclidean k-means Recently Ahmadian et al. [ANFSW17] improved the approximation ratio given by [KMN+04] to $(6.357+\varepsilon)$ for the discrete k-means problem. There have been works to get a PTAS for Euclidean k-means objective. In order to obtain the PTAS, many have focused on cases where k or d or both are assumed to be fixed. Inaba et al. [IKI94] gave a PTAS when both k and d is fixed. There have been a series of work in the case when only k is assumed to be fixed [VKKR03, HPM04, HPK05, FMS07, KSS04, Che09]. Recently there has been works which give a PTAS for Euclidean k-means where only d is assumed to be a constant [CAKM16, FRS16]. Charikar et al. [CGTS02] gave the first constant factor approximation ratio of $6\frac{2}{3}$ to the discrete k-median problem. Then, Jain and Vazirani [JV01] gave a 6-approximation algorithm for the same. Li and Svensson [LS13] showed a $(1+\sqrt{3}+\varepsilon)$-approximation algorithm for the discrete k-median. The best known approximation for the discrete k-median problem is 2.675-factor approximation algorithm by Bryka et al. [BPR+17] .

ℓ_p Cost k-Clustering For $p = 1$ the problem is known as the k-median clustering. For $p = 2$ the clustering corresponds to the k-means clustering and the clustering cost corresponds to $\sqrt{k\text{-means cost}}$, which are well-studied problems. For general $p \geqslant 1$ the problem is also known as the ℓ_p-norm minimization in the k-clustering setting. Chakrabarty and Swamy [CS19] show a $O(C)$-factor approximation algorithm for the discrete ℓ_p-norm minimization in the k-clustering setting problem, for some positive constant C.

Sampling Based Methods Our results use sampling techniques often used in $(1+\varepsilon)$- approximation for k-means [KSS04, Che09, ABS10]. The first ever linear (in n and d) running time for obtaining PTAS (assuming k to be a constant) given by [KSS04] is $O\left(nd2^{poly(k/\varepsilon)}\right)$. Feldman et al. [FMS07] gave a new algorithm (using efficient coreset construction) with a better running time than that of [KSS04] from $O\left(nd2^{poly(k/\varepsilon)}\right)$ to $O\left(nkd + d.poly(k/\varepsilon) + 2^{\tilde{O}(k/\varepsilon)}\right)$. There have been other works which also show similar results using D^2 sampling method [JKS14, BJK18]. Ding and Xu [DX15] gave a sampling based procedure to cluster other variants of k-means objective, which they called the constrained k-means clustering. These clustering objectives need not satisfy the locality property in Euclidean space. Their algorithm is based on uniform sampling and some stand alone geometric technique which they call the 'simplex lemma'. The running time of their algorithm is $O\left(2^{\text{poly}(k/\varepsilon)}n(\log n)^{k+1}d\right)$. Deshpande et al. [DLS19] use these techniques to exactly solve the k-means clustering for balanced clusters of α-center proximal instances in time $O\left(2^{\text{poly}(k/\varepsilon)}nd\right)$. If the pairwise distance between the means is within a factor of gamma of each other, then they show an exact algorithm for minimizing the k-means objective over clustering that satisfy α-center proximity and the running time of their algorithm depends exponentially on the factor γ, number of clusters k, and linear in n and d. They also show an exact algorithm for minimizing the k-means objective over clustering that satisfy α-center and form balanced clusters. The running time depends exponentially on the balance parameter $1/\omega$, number of clusters k, and linear in n and d where the size of each cluster is at least $\omega n/k$. Bhattacharya et al. [BJK18] gave a more efficient algorithm based on D^2 sampling for the same class of constrained k-means problem studied by [DX15] and gave an algorithm with running time $O\left(2^{\tilde{O}(k/\varepsilon)}nd\right)$. All the work related to sampling based methods above estimate the means/centers of the clusters, and use them to recover a clustering. Bhattacharya et al. [BJK18] also show an extension of the constrained k-means techniques to the constrained k-median setting.

Cost-Balanced Problems The discrete versions of cost-balanced clustering problems in abstract metric space has been studied. It is studied under different names, like Minimum load k-facility location (MLkFL), and min-max star cover. MLkFL is defined as follows: given a set of facilities and clients and an integer $k > 0$, the goal is to open a subset of k facilities and assign clients to the facilities, such that the load of the heaviest facility is minimized, where the load of a facility is sum of distances of clients, from the facility, assigned to it. Evan et al. [EGK+03] and Arkin et al. [AHL06] studied the cost-balanced clustering problem under the name of min-max star cover. They view the problem as one where we try to cover the the nodes of a graph by stars. The goal is to come up with k stars such that the cost of the heaviest star is minimized, where the cost of a star is measured as the sum of distances of nodes of the star from its root. Evan et al. [EGK+03] gave a $(4,4)$ bi-criteria approximation for the problem. This was improved by Arkin et al. [AHL06] who gave a $(3+\varepsilon, 3+\varepsilon)$ bi-criteria approximation for the same. Evan et al. [EGK+03] and Arkin et al. [AHL06], consider other problems where they try to cover the graph with different objects like trees, paths, etc. Ahmadian et al. [ABF+18] look at the same problem under the name of minimum-load k-facility location (MLkFL), and gave a PTAS on line metrics. They also give a 12 factor approximation algorithm for MLkFL on star metrics. On the hardness side they show that MLkFL is strongly NP-Hard on line metrics; they also show that even a configuration-style LP-relaxation has a bad integrality gap, and a multi-swap k-median style local-search heuristic has a bad locality gap.

Makespan Scheduling Minimum makespan scheduling on related machines is a well studied problem. There is a factor 2 approximation factor known for it [L.96] and a PTAS by [HS87]. The problem is known to be strongly NP-Hard [GJ90]. Minimum makespan scheduling on unrelated parallel machines NP hard to approximate to a factor better than $3/2 - \varepsilon$ [LST90]. Lenstra et al. [LST90] gave a factor 2 approximation algorithm for the problem, which is currently best know approximation achieved in polynomial time. Horowitz and Sahani [HS76] gave a polynomial time algorithm where the number of machines k is a fixed constant that computes a $(1+\varepsilon)$ approximation in time $O\left(nk(nk/\varepsilon)^{k-1}\right)$ for any $\varepsilon > 0$, where n is the number of jobs. Fishkin et al. [FJM08] improved the running time of [HS76] (there were works in between also like [JP01]) by giving a $(1+\varepsilon)$ approximation scheme whose running time is $O(n) + (\log m/\varepsilon)^{O(m^2)}$. Jansen and Mastrolilli [JM10] gave a $(1+\varepsilon)$ approximation scheme whose running time is $O(n) + 2^{O(m\log(m/\varepsilon))}$.

1.3 Proof Overview

One way to go about clustering a set of points is to first estimate the k cluster centers, and then "assign" the points to these centers. This approach has been studied in the context of k-means clustering [Che09, FMS07, HPK05, HPM04, KSS04, VKKR03], and other objective functions. It is a well known that for the k-means objective function, if the optimal cluster centers are known, then assigning each point to the nearest center (breaking ties arbitrarily) will recover the optimal cost clustering. However, for the cost-balanced k-means objective, an analogous statement is not true. In fact, given a set of centers $c_1, \ldots, c_k$, assigning the points optimally to these centers is an NP-hard problem (Proposition 1.13). We show that, given a set of centers $c_1, \ldots, c_k$, we can use an approximation algorithm for scheduling jobs on unrelated machines (due to [JM10]), to compute a $(1+\varepsilon)$-approximation to the optimal assignment of points to $c_1, \ldots, c_k$ (Theorem 2.5). Next, we show that to obtain our result (Theorem 1.6), it suffices to estimate the cluster centers to a sufficiently high accuracy. We use the algorithm of [BJK18] to estimate the cluster centers in our setting; their analysis of their algorithm, with appropriate modifications wherever needed, works for our setting. The analysis of Theorem 1.7 follows in a similar fashion by using a lemma of [KSS10] (Lemma C.1) which gives a procedure to approximately find the 1-median center of a cluster, given random samples from that cluster.

To prove Theorem 1.8, we use the idea that the optimal ℓ_p cost k-clustering cost is upper bounded by k times the optimal ℓ_p cost-balanced k-clustering cost. This also implies that an α-approximation to the optimal ℓ_p cost k-clustering cost is upper bounded by $\alpha \cdot k$ times the optimal ℓ_p cost-balanced k-clustering cost. In the theorem we assume access to a black-box algorithm, which gives a clustering with their corresponding centers, that give an α-approximation to the optimal ℓ_p cost k-clustering cost. Suppose, we also knew the OPT, which is the cost of the optimal ℓ_p cost-balanced k-clustering clustering. Then, using the given ℓ_p cost k-clustering clustering by a black-box α-approximation algorithm, we do a bin-packing type analysis, where we greedily assign points to the given centers till we exceed the $O\left(\alpha\mathsf{OPT}/\varepsilon^{1/p}\right)$ cost of the clustering with the center. We then proceed to make a copy of the centers for which we exceeded the $O\left(\alpha\mathsf{OPT}/\varepsilon^{1/p}\right)$ cost and repeat the above procedure, till all the clusters are of cost at most $O\left(\alpha\mathsf{OPT}/\varepsilon^{1/p}\right)$. We show that we would need to open a maximum of $(1+\varepsilon)k$ clusters. This implies a $\left(O\left(\alpha/\varepsilon^{1/p}\right), (1+\varepsilon)\right)$ factor bi-criteria approximation to the optimal ℓ_p cost-balanced k-clustering cost.

We also mention in Appendix A that, for the discrete version of the cost-balanced k-means the natural facility-location type linear programming approach cannot be used to give a constant factor approximation algorithm. To this end, we mention the integrality gap instance given by [Cha18], which gives us an integrality gap of $(k+1)$.

Notations: The mean of a set of finite set of points $X \subset \mathbb{R}^d$ is denoted by $\mu(X)$. Let $\Delta(X)$ denote the 1-means cost of these set of points, i.e., $\Delta(X) \stackrel{\text{def}}{=} \sum_{x\in X} \|x-\mu(X)\|^2$. A k-partition of X into disjoint subsets $\mathbb{O} = \{O_1, \ldots, O_k\}$ is called a k-clustering of X. We denote the optimal cost-balanced k-means clustering by $\mathbb{O}^* = \left\{O_1^*, \ldots, O_k^*\right\}$. Given a clustering $\mathbb{O}$ and a set $C = \{c_1, \ldots, c_k\}$, we define $cost_C(\mathbb{O})$ as the minimum over all permutation π of C of $\max_{i\in[k]} \sum_{x\in O_i} \left\|x - c_{\pi(i)}\right\|^2$. Recall that OPT denotes the optimal value of the cost-balanced k-means ($\mathcal{K}(\mathbb{O}^*)$). For a set of points X and another set of points C, we define $\phi_C(X) = \sum_{x\in X} \min_{c\in C} \|x-c\|^2$. With a slight abuse of notation, when set C has only one element c, we will use the notation $\phi_c(X)$, instead of $\phi_{\{c\}}(X)$.

2 $(1+\varepsilon)$-Approximation for Cost-Balanced k-Means Clustering

We define the notion of D^2-sampling, which will be used by the Algorithm 2

Definition 2.1 (D^2-sampling)**.** Given a set of points $X \subset \mathbb{R}^d$ and another set of points $C \subset \mathbb{R}^d$, D^2-sampling from X w.r.t. C samples a point $x \in X$ with probability $\frac{\phi_C(\{x\})}{\phi_C(X)}$. When $C = \emptyset$, we pick a point uniformly at random from X.

In the following theorem, we prove that Algorithm 2 gives us a set of k points (centers) such that, with constant probability, the cost of the optimal clustering with respect to these centers is at most $(1+\varepsilon)$ times the optimal cost of cost-balanced k-means .

Theorem 2.2. *Algorithm 1 takes input a set of points $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$, parameters k, ε, and constructs a list $\mathcal{L}$ of $2^{\tilde{O}(k/\varepsilon)}$ sets of centers of size k such that for an optimal cost-balanced k-means clustering $\mathbb{O}^* = \left\{O_1^*, \ldots, O_k^*\right\}$ of X, the*

Algorithm 1: Cost-Balanced k-Means Algorithm

Input: Set of points $X \subset \mathbb{R}^d$, number of clusters k, and an error parameter ε.
Output: A cost-balanced k-means clustering $\mathbb{O}^A$.

1. Let $N = \frac{136448\ k}{\varepsilon^3}$, $M = \frac{100}{\varepsilon}$.
2. Initialize $\mathcal{L}$ to $\emptyset$. $\mathcal{L}$ will contain a list of candidate means of a clustering, where each candidate mean is a set of *exactly* k centers.
3. Repeat 2^k times:
   - Make a call to (Algorithm 2) Sample-Centers$(X, k, \varepsilon, \mathcal{L}, 0, \{\})$.
4. For each tuple t in $\mathcal{L}$:
   - Form a matrix $\mathcal{J}_{[k \times n]}$, where $\mathcal{J}(i, j) = \left\| t_i - x_j \right\|^2$.
   - Input $t, \mathcal{J}$ to the Jansen & Mastrolilli's [JM10] algorithm for minimum makespan scheduling on unrelated machines.
   - Maintain the clustering with the minimum cost.
5. Return the minimum cost clustering $\mathbb{O}^A$.

Algorithm 2: Sample-Centers Algorithm (Subroutine of Algorithm 2.1, [BJK18])

Input: Set of points $X \subset \mathbb{R}^d$, number of clusters k, an error parameter ε, a list $\mathcal{L}$ of k-tuples, index i , and a set C of centers.

1. Set $N = \frac{136448\ k}{\varepsilon^3}$, $M = \frac{100}{\varepsilon}$, $S' = \emptyset$.
2. If $(i = k)$ then add C to the set $\mathcal{L}$.
3. else
   (a) S is an i.i.d. sample of N points picked by D^2-sampling (Definition 2.1) w.r.t. C.
   (b) $S' \leftarrow S$.
   (c) For all $c \in C : S' \leftarrow S' \cup \{M \text{ copies of } c\}$.
   (d) For all subsets T which is a collection of M points from S' (with repetitions allowed):
      i. $C \leftarrow C \cup \{\mu(T)\}$, where $\mu(T)$ is the means of points in the set T.
      ii. Sample-Centers$(X, k, \varepsilon, \mathcal{L}, i + 1, C)$.

following event happens with probability at least $1/2$*: there exists a set* $C \in \mathcal{L}$ *such that*

$$cost_C(\mathbb{O}^*) \leqslant (1+\varepsilon)\,\mathsf{OPT}\,.$$

Moreover, the running time of the algorithm is $O\left(nd\, 2^{\tilde{O}(k/\varepsilon)}\right)$.

We will prove Theorem 2.2 in Section B. Theorem 2.2 gives us a way to estimate the centers of a good clustering (i.e., a clustering whose cost-balanced k-means cost is at most $1+\varepsilon$ times OPT). However, recovering a good clustering by "assigning" points to these centers is a non-trivial problem; Proposition 1.13 shows that even if the centers of the optimal clustering are known, recovering the optimal clustering from it is an NP-hard problem). We reduce the problem of assigning the points to the cluster centers, to the problem of scheduling jobs on machines so as to minimize the makespan. More formally, the scheduling problem is the following.

Problem 2.3. There are k parallel machines and n independent jobs. Each job is to be assigned to one of the machines. The processing of job j on machine i requires time $p_{i,j}$. The objective is to find a schedule that minimizes the makespan (the total time that elapses from the beginning to the end).

The problem of assigning points to the cluster centers can be viewed as a special case of Problem 2.3. This follows from the following lemma.

Lemma 2.4. *Given a set of* n *points* $X = \{x_1, ..., x_n\}$ *and a set* $C = \{c_1, ..., c_k\}$ *of* k *points in* $\mathbb{R}^d$*, one can construct an instance of minimizing the makespan of scheduling on unrelated parallel machines, the cost of which is the same as the cost of a clustering* $\mathbb{O}^* = \left\{O_1^*, ..., O_k^*\right\}$ *which minimizes the* $cost_C(\mathbb{O}^*)$.

Proof. We use the index i for the set of machines and j for the jobs. We treat the n points X as n jobs and the set C of k points as k unrelated parallel machines. For a job j, the processing on machine i is the following: $p_{i,j} = \left\|x_j - c_i\right\|^2$. We note that a schedule corresponds to a clustering: since each job is assigned only to one machine, and all the jobs are processed. The makespan of a schedule is equal to the sum of processing times assigned to the machine with the largest "load".

Given an optimal solution of minimizing the makespan of scheduling on unrelated parallel machines to the instance constructed above, one can find an optimal clustering $\mathbb{O}^* = \left\{O_1^*, ..., O_k^*\right\}$ which minimizes the $cost_C(\mathbb{O}^*)$ as follows: if a job j is scheduled on a machine i, then x_j is put in the O_i^* cluster ($x_j \in O_i^*$). By construction, the makespan of the schedule corresponds to $cost_C(\mathbb{O}^*)$. □

However, as mentioned above, given a set X of n points and a set C of k points it is NP-Hard to find an optimal clustering $\mathbb{O}^*$ which minimizes the $cost_C(\mathbb{O}^*)$. We prove this in the Proposition 1.13.

Jansen and Mastrolilli [JM10] gave a $(1+\varepsilon)$ approximation algorithm for the minimizing the makespan of scheduling n jobs on k unrelated parallel machines, which runs in $O(n) + 2^{O(k\log(1/\varepsilon))}$.

Theorem 2.5 ([JM10])**.** *There is an* $(1+\varepsilon)$*-approximation algorithm for the non-preemptive minimum makespan problem with* n *jobs and* k *unrelated parallel machines that runs in* $O(n) + 2^{\tilde{O}(k/\varepsilon)}$.

Proof of Theorem 1.6. After the Step 3 of Algorithm 1, the algorithm constructs a list $\mathcal{L}$ of the candidate centers (recall that the $\mathcal{L}$ is a is a list of candidate centers for a clustering, where each candidate center is a set of exactly k points), the optimal assignment to one of which gives a $(1+\xi)$-approximation to the optimal cost-balanced k-means. This is guaranteed by Theorem 2.2. The Step 4 of Algorithm 1 finds a clustering (using the reduction from the Lemma 2.4) for all the k-tuple t in $\mathcal{L}$. For at least one $t \in \mathcal{L}$ we get an assignment to the given candidate centers t, which is a $(1+\xi)$-approximation to optimal assignment to the candidate set of centers. Therefore, denoting the optimal clustering obtained by the set of centers C as $\mathbb{O}^C$ and the clustering obtained by our algorithm as $\mathbb{O}^A$, we get the following inequalities:

$$\mathcal{K}\left(\mathbb{O}^A\right) \leqslant cost_C(\mathbb{O}^A) \leqslant (1+\xi)cost_C(\mathbb{O}^C) \leqslant (1+\xi)^2\mathsf{OPT}\,,$$

where the first inequality is direct as $\mathcal{K}\left(\mathbb{O}^A\right)$ is the cost-balanced k-means cost of the clustering $\mathbb{O}^A$, and $cost_C(\mathbb{O}^A)$ is the cost, when the centers of a cluster are constrained to be in the set C. From Theorem 2.5 we get that $cost_C(\mathbb{O}^A) \leqslant (1+\xi)cost_C(\mathbb{O}^C)$. From Theorem 2.2 we get that $(1+\xi)cost_C(\mathbb{O}^C) \leqslant (1+\xi)^2\mathsf{OPT}$.

Therefore, setting $\varepsilon = \xi/3$, we get that

$$\mathcal{K}\left(\mathbb{O}^{A}\right) \leqslant (1+\varepsilon)\mathsf{OPT}\,.$$

Run-time Analysis: The size of the list $\mathcal{L}$ produced by Step 3 of Algorithm 1 is $2^{\tilde{O}(k/\varepsilon)}$. The time required to obtain such a list is $O\left(nd\, 2^{\tilde{O}(k/\varepsilon)}\right)$. The step 4 of Algorithm 1 uses the Algorithm by Jansen and Mastrolilli [JM10] for each set of centers t. The algorithm simply iterates over the list $\mathcal{L}$ and use the algorithm given by Jansen and Mastrolilli [JM10] and we obtain a running time of:

$$\left(O\left(nd\right) + 2^{\tilde{O}(k/\varepsilon)}\right)2^{\tilde{O}(k/\varepsilon)} + O\left(nd\, 2^{\tilde{O}(k/\varepsilon)}\right) = O\left(nd\, 2^{\tilde{O}(k/\varepsilon)}\right).$$

The success probability of the algorithm is determined by Theorem 2.2, which is a constant. □

3 Bi-Criteria Approximation

Given a metric space (X, dist), where X represents the set of points and $\mathrm{dist}(u,v)$ is the distance between $u, v \in X$. Suppose we had access to an algorithm which would a α-factor approximation to the following *ℓ_p cost k-clustering* objective: Find a k-partitioning of the points into $O_1, ..., O_k$, and corresponding centers $c_1, ..., c_k \in X$ such that $\left(\sum_{i=1}^{k}\sum_{u\in O_i}\mathrm{dist}(u,c_i)^p\right)^{\frac{1}{p}}$ is minimized, where $p \in \mathbb{R}_+$. For $p = 1$, the problem becomes k-median.

In this section, we will show a procedure that would give a bi-criteria $(\alpha(2(1+\xi)/\varepsilon)^{\frac{1}{p}}, (1+\varepsilon))$, where $\xi > 0, \varepsilon > 0$ are parameters to the algorithm, approximation of the *ℓ_p cost-balnced k-clustering* problem, i.e., find a k-partitioning of the points into $O_1, ..., O_k$, with corresponding centers $c_1, ..., c_k \in X$ such that $\left(\max_{i\in[k]}\sum_{u\in O_i}\mathrm{dist}(u,c_i)^p\right)^{\frac{1}{p}}$ is minimized, for $p \in \mathbb{R}_+$, in time $\mathsf{poly}\left(n, k, p, \frac{1}{\log(1+\xi)}\right)$.

Our procedure would partition the points into $(1+\varepsilon)k$ clusters and give a $\left((2((1+\xi)/\varepsilon)^{\frac{1}{p}}\alpha\right)$-factor approximation to the objective.

Notations: Let OPT be the optimal ℓ_p cost-balnced k-clustering cost and let OPT_Σ be the optimal ℓ_p cost k-clustering cost. Let $\mathcal{G}$ be the α-approximation algorithm for ℓ_p cost k-clustering problem, which outputs the clusters as $\mathbb{O} = \{O_1, \ldots, O_k\}$ with corresponding centers as $\mathbb{C} = \{c_1, ..., c_k\}$. Let $\Delta(O_i, c_i) = \sum_{x\in O_i}\mathrm{dist}\,(x, c_i)^p$ be ℓ_p^p assignment cost for cluster O_i.

Algorithm 3: Bi-Criteria approximation for ℓ_p cost-balanced k-clustering problem

Input: Metric space (X, dist), number of clusters k, error parameters ε and ξ, and an α-approximation algorithm $\mathcal{G}$ for the ℓ_p cost k-clustering problem.

Output: $(1+\varepsilon)k$ clusters $\mathbb{O}^{\mathcal{A}}$, and centers $\mathbb{C}^{\mathcal{A}}$ such that $\mathcal{K}_p(\mathbb{O}^{\mathcal{A}}, \mathbb{C}^{\mathcal{A}}) = \left(\frac{2(1+\xi)}{\varepsilon}\right)^{\frac{1}{p}}\alpha\mathsf{OPT}$.

1. $\mathbb{O}^{\mathcal{A}}, \mathbb{C}^{\mathcal{A}} = \emptyset$
2. Run $\mathcal{G}$ and obtain a k-partition of X into $\mathbb{O} = \{O_1, \ldots, O_k\}$ with corresponding centers as $\mathbb{C} = \{c_1, ..., c_k\}$.
3. Guess the value of OPT^p within a factor of $(1+\xi)$ and let that value be denoted by OPT_g.
4. For each O_i, $i \in [k]$, do
   (a) Make a separate cluster for each of the points in $x \in O_i$ with $\mathrm{dist}\,(x, c_i)^p \geqslant \frac{2\alpha^p\mathsf{OPT}_g}{\varepsilon}$ and add each of them to $\mathbb{O}^{\mathcal{A}}$ and $\mathbb{C}^{\mathcal{A}}$
   (b) If O_i is not empty, assign points from O_i to c_i in a greedy manner until total cost doesn't exceed $\frac{2\alpha^p\mathsf{OPT}_g}{\varepsilon}$ and add this cluster to $\mathbb{O}^{\mathcal{A}}$ and c_i to $\mathbb{C}^{\mathcal{A}}$. Remove all assigned points from O_i.
   (c) Go to 4(b) again if O_i is not empty.

Lemma 3.1. *If the guess* OPT_g *in step* 3 *of Algorithm 3 is between* OPT^p *and* $(1+\xi)\mathsf{OPT}^p$*, then the number of centers opened by Algorithm 3 is at most* $(1+\varepsilon)k$*, and the* ℓ_p^p *assignment cost for each cluster is at most* $\left(\frac{2(1+\xi)}{\varepsilon}\right)\alpha^p\mathsf{OPT}^p$*. The running time of the algorithm is* $O\left(\frac{n(\log(k)+p\log(\alpha))}{\log(1+\xi)}\right)$ + *RunningTime*$(\mathcal{G})$.

Proof. We know that $\mathsf{OPT}_\Sigma^p \leqslant k\mathsf{OPT}^p$ and solution returned by $\mathcal{G}$ gives a α-factor approximation to OPT_Σ. Therefore, the following equation holds:

$$\sum_{i=1}^{k}\sum_{x\in O_i}\operatorname{dist}(x,c_i)^p \leqslant k\alpha^p\mathsf{OPT}^p\,. \tag{3}$$

Step 4 of the algorithm partitions each O_i, $i\in[k]$, into smaller clusters (with their corresponding center), each with the cost of at most $\frac{2\alpha^p\mathsf{OPT}_g}{\varepsilon}$.

Let $S\subseteq X$ be set of points selected by step 4(a) of the algorithm,

$$\sum_{i=1}^{k}\Delta(S\cap O_i,c_i)=\sum_{i=1}^{k}\sum_{x\in S\cap O_i}\operatorname{dist}(x,c_i)^p \geqslant \frac{2\alpha^p\mathsf{OPT}_g\,|S|}{\varepsilon} \tag{4}$$

Since $\sum_{i=1}^{k}\Delta(O_i,c_i)=\sum_{i=1}^{k}(\Delta(O_i\setminus S,c_i)+\Delta(S\cap O_i,c_i))$,

$$\sum_{i=1}^{k}\Delta(O_i\setminus S,c_i)\leqslant\sum_{i=1}^{k}\Delta(O_i,c_i)-\frac{2\alpha^p\mathsf{OPT}_g\,|S|}{\varepsilon} \tag{5}$$

The inequality in above equation follows from inequality of (4).

For each i, step 4(b) could be thought as bin packing problem where items corresponds to clients in $O_i\setminus S$ and capacity of bin is $\frac{2\alpha^p\mathsf{OPT}_g}{\varepsilon}$. Each point $x\in O_i\setminus S$ could be thought of as item having weight $\operatorname{dist}(x,c_i)^p$. The sum of weight of corresponding bin packing instance for cluster O_i is at most $\Delta(O_i\setminus S,c_i)$.

If bin packing instance have total size of items W and capacity of each bin is w then, first fit bin packing algorithm opens at most $\left(2\frac{W}{w}+1\right)$ bins. This result has been proved in Problem 9.1 of chapter 9 of [Vaz03].

By using the above fact, step 4(b) of the algorithm will partition $O_i\setminus S$ into at most $2\frac{\varepsilon\Delta(O_i\setminus S,c_i)}{2\alpha^p\mathsf{OPT}_g}+1$. So the total number of clusters opened by step 4(b) of the algorithm Algorithm 3 is at most

$$\begin{aligned}
2\sum_{i=1}^{k}\frac{\varepsilon\Delta(O_i\setminus S,c_i)}{2\alpha^p\mathsf{OPT}_g}+k &= 2\left(\frac{\varepsilon\sum_{i=1}^{k}\Delta(O_i\setminus S,c_i)}{2\alpha^p\mathsf{OPT}_g}\right)+k\\
&\leqslant 2\left(\frac{\varepsilon\sum_{i=1}^{k}\Delta(O_i,c_i)}{2\alpha^p\mathsf{OPT}_g}\right)-2\,|S|+k\\
&\leqslant (1+\varepsilon)k-2\,|S|\,.
\end{aligned}$$

Second inequality follows from (5) and third inequality in above equation follows from (3) and our assumption that $\mathsf{OPT}^p\leqslant\mathsf{OPT}_g$.

Total number of clusters opened by Algorithm 3 is sum of number of clusters opened by step 4(a) and 4(b) which is at most $(1+\varepsilon)k-|S|\leqslant(1+\varepsilon)k$.

Step 4(a) ensures that each cluster returned by algorithm has ℓ_p^p assignment cost at most $\left(\frac{2(1+\xi)}{\varepsilon}\right)\alpha^p\mathsf{OPT}^p$.

Run-Time Analysis: Let the $\Delta_p(\mathcal{G})=\Delta_p(\mathbb{O},\mathbb{C})$. We know that $\frac{\Delta_p^p(\mathcal{G})}{k\alpha^p}\leqslant\mathsf{OPT}^p\leqslant\Delta_p^p(\mathcal{G})$, since $\mathsf{OPT}_\Sigma^p\leqslant k\mathsf{OPT}^p$ and $\Delta_p^p(\mathcal{G})\leqslant\alpha^p\mathsf{OPT}_\Sigma^p$. Therefore, the maximum number of iterations needed in the step 3 of the algorithm to get within a factor of $(1+\xi)$ to OPT^p is at most $\frac{\log(k)+p\log(\alpha)}{\log(1+\xi)}$. The step 4 takes time linear in the number of points n. □

Remark 3.2. Note that the set of centers returned by Algorithm 3 depends on set of centers returned by $\mathcal{G}$. If $\mathcal{G}$ is the discrete version of the ℓ_p cost k-clustering, then the set of centers returned by $\mathcal{G}$ belongs to X. If $\mathcal{G}$ is the Euclidean version of the ℓ_p cost k-clustering, then the set of centers returned by $\mathcal{G}$ belongs to $\mathbb{R}^d$.

Proof of Theorem 1.8. The theorem follows directly from the above Lemma 3.1. □

4 Lower Bound

Proof of Proposition 1.13. The minimum makespan scheduling on identical machines is defined as follows Given processing times for n jobs, $p_1, p_2, ..., p_n$, and an integer k, find an assignment of the jobs to k identical machines so that the completion time, also called the makespan, is minimized. The problem of minimum makespan scheduling (on identical machines) is NP-hard. The problem is NP-hard, even if there are only two identical machines [GJ90]. This problem can be seen as a special case of Problem 2.3.

Suppose we are given an instance of minimum makespan scheduling with k machines and n jobs, and a processing time of p_j for each job $j \in [n]$. We will construct an instance of cost-balanced k-means problem with a given set of centers. For each machine $i \in [k]$ we assign a zero vector in $\mathbb{R}$ (on a line). For each job j, we look at the value at the running time p_j and set $x_j = \sqrt{p_j}$ on the line. Set set $\{x_1, \ldots, x_n\}$ form our instance of cost-balanced k-means problem with a given set of centers.

If we can solve cost-balanced k-means problem with a given set of centers optimally in polynomial time, then we can solve the problem of minimum makespan scheduling (on identical machines) in polynomial time (by construction, as the cost of assigning a point x_i to a cluster is equivalent to the running time of job i, which is p_i). Therefore, the proof is complete by contrapositive of the above statement. □

Acknowledgements

AL is grateful to Microsoft Research for supporting this collaboration. AL would like to thank Moses Charikar and Paris Syminelakis for helpful discussions. AL was supported in part by SERB Award ECR/2017/003296. The authors would like to thank anonymous reviewers for their valuable feedback.

A LP Relaxation and Integrality Gap Instance Due to [Cha18]

We try to look at the discrete version of cost-balanced k-means , where the cluster centers are constrained to come from the set of given points. One can try to approach such a problem via integer linear programming (ILP), similar to facility location problems. A natural ILP for the discrete version of the cost-balanced k-means would be as follows. For convenience, let us denote $\mathcal{F}$ as the set of facilities, and C as the set of clients. In our case, $C = \mathcal{F} = X$. Let the distance between the i^{th} facility and the j^{th} location be denoted by $d(i, j)$. We define the variables y_i for $i \in \{1, ..., |X|\}$, and x_{ij} for $i \in \{1, ..., |X|\}$ and $j \in \{1, ..., |X|\}$.

$$
\begin{aligned}
\text{minimize} \quad & \alpha \\
\text{subjected to} \quad & \sum_{i \in \mathcal{F}} x_{ij} = 1, && \forall j \in C; \\
& \sum_{i \in \mathcal{F}} y_i \leqslant k; \\
& x_{ij} \leqslant y_i, && \forall i \in \mathcal{F}, j \in C; \\
& \sum_{j \in C} d(i, j)^2 \, x_{ij} \leqslant \alpha, && \forall i \in \mathcal{F}; \\
& y_i, x_{ij} \in \{0, 1\}, && \forall i \in \mathcal{F}, j \in C.
\end{aligned}
$$

We can relax the last constraint, such that the variables $x_{ij}, y_i \in [0, 1]$. We show that the relaxed linear program has an integrality gap of $k + 1$. The constraints of the above ILP are a standard set of constraints, which are used for analyzing facility location problems; with an additional constraint, that the cost of each cluster should be less that some quantity α. Therefore, the objective is to minimize the α.

The following integrality gap instance was pointed to us by Charikar [Cha18]. Consider the following instance, with $k + 1$ points, i.e., $|X| = k + 1$. The points are at a distance of 1 from each other, i.e., $d(i, j) = 1$ for $i \neq j$ and $d(i, i) = 0$. The optimal discrete cost-balanced k-means solution for such an instance would show up if we choose any k points as centers. The cost for such k points is 0. For the $(k + 1)^{th}$ point, we assign it to any one of the k points. This yields a

discrete cost-balanced k-means cost of 1. The relaxed linear programming solution is $y_i = 1 - \frac{1}{k+1}$, and $x_{ii} = 1 - \frac{1}{k+1}$, and $x_{ij} = \frac{1}{k(k+1)}$, for all $i \neq j \in X$. Whence we evaluate the cost of the optimal fractional linear program, we get that cost is $1/(k+1)$. Therefore, this yields an integrality gap of greater than $(k+1)$.

B Finding Good Candidate Centers

The following fact is commonly known as the parallel axis theorem. We will state it as a fact.

Fact B.1. *For any $X \subset \mathbb{R}^d$ and $c \in \mathbb{R}^d$, we have*

$$\sum_{x \in X} \|x - c\|^2 = \sum_{x \in X} \|x - \mu(X)\|^2 + |X| \, \|c - \mu(X)\|^2 \, .$$

We will also use a triangle inequality type inequality for squared Euclidean norm. We call it the approximate triangle inequality.

Fact B.2 (Approximate Triangle Inequality)**.** *For any $x, y, z \in \mathbb{R}^d$, we have*

$$\|x - z\|^2 \leqslant 2\left(\|x - y\|^2 + \|y - z\|^2\right) .$$

We will use the following result which shows that to estimate the 1-means cost of a set of points (in our case, one the k clusters), it suffices to estimate its mean using a random sample of points from it.

Lemma B.3 ([IKI94])**.** *Let S be a set of points obtained by i.i.d. uniformly sampling M points from a point set $X \subset \mathbb{R}^d$. Then for any $\delta > 0$,*

$$\mathbb{P}\left[\phi_{\mu(S)}(X) \leqslant \left(1 + \frac{1}{\delta M}\right) \cdot \Delta(X)\right] \geqslant (1 - \delta) \, .$$

Proof of Theorem 2.2. The proof of Theorem 2.2 is similar to the Theorem 1 of [BJK18], with minor modifications. They consider the k-means objective, where the cost of a k-clustering is the sum of all the k clusters. In our case (cost-balanced k-means), we have that the cost of a k-clustering is the maximum of the individual k clusters. Therefore using the analysis of [BJK18] and fact that the sum of cost of the k clusters is at most k times the cost of the maximum cluster, we obtain the statement of Theorem 2.2. We reproduce many statements from [BJK18] with nearly identical proofs but cannot simply use the lemmas from [BJK18] as black-boxes because the OPT in our paper is the optimum of the cost-balanced k-means and not the usual 'min-sum' type objectives as in [BJK18].

It will be useful to think of the execution of this algorithm as a tree $\mathcal{T}$ of depth k. Each node in the tree can be labeled with a set C, it corresponds to the invocation of Algorithm 2 with this set as C (and i being the depth of this node). The children of a node denote the recursive function calls by the corresponding invocation of Algorithm 2. Finally, the leaves denote the set of candidate centers constructed by the algorithm.

We will argue that the following invariant $P(i)$ is maintained during the recursive calls to the Sample-centers Algorithm 2:

***P*(*i*) :** With probability at least $\frac{1}{2^{i-1}}$, there is a node v_i at depth $(i-1)$ in the tree $\mathcal{T}$ and a set of $(i-1)$ distinct clusters $O^*_{j_1}, O^*_{j_2}, \ldots, O^*_{j_{i-1}}$ such that

$$\forall l \in \{1, \ldots, i-1\} \, , \phi_{c_l}(O^*_{j_l}) \leqslant \left(1 + \frac{\varepsilon}{2}\right) \Delta(O^*_{j_l}) + \frac{\varepsilon}{2} \mathsf{OPT} \, , \tag{6}$$

where $c_1, \ldots, c_{i-1}$ are the centers in the set C_{v_i} corresponding to v_i.

We prove this via induction.

Base Case: The base case for $i = 1$ follows trivially: the vertex v_1 is the root of the tree $\mathcal{T}$ and C_{v_1} is empty.

Induction Step: We now assume that $P(i)$ holds for some $i \geqslant 1$. We will prove that $P(i+1)$ also holds. For ease of notation, without loss of generality, we assume that the index j_i is i, and use C_i to denote C_{v_i}. Thus, the center c_l corresponds to O^*_l, $1 \leqslant l \leqslant i-1$.

We will use the index i' to represent a new un-sampled cluster $i' \geqslant i$. Note that $\phi_{C_i}(O^*_{i'})$ is proportional to the probability that a point sampled from X using D^2-sampling w.r.t. centers C_i comes from $O^*_{i'}$. Let $\bar{i}$ be the index i' for

which $\phi_{C_i}(O^*_{i'})$ is the maximum. We will argue that the invocation of sample centers in step $3(d)(i)$ will consider a point c_i, such that the following property holds with probability at least $1/2$:

$$\phi_{c_i}(O^*_i) \leqslant \left(1 + \frac{\varepsilon}{2}\right)\Delta(O^*_i) + \frac{\varepsilon}{2}\mathsf{OPT}\,.$$

We break the analysis into two parts:

$$\textbf{Case 1:}\ \left(\frac{\phi_{C_i}(O^*_i)}{\sum_{j=1}^k \phi_{C_i}(O^*_j)} < \frac{\varepsilon}{13k}\right) \qquad \& \qquad \textbf{Case 2:}\ \left(\frac{\phi_{C_i}(O^*_i)}{\sum_{j=1}^k \phi_{C_i}(O^*_j)} \geqslant \frac{\varepsilon}{13k}\right).$$

The condition of these two cases are identical as given in [BJK18].

Case 1 $\left(\frac{\phi_{C_i}(O^*_i)}{\sum_{j=1}^k \phi_{C_i}(O^*_j)} < \frac{\varepsilon}{13k}\right)$**:** This captures the scenario where the probability of sampling from uncovered clusters is very small. In this case we argue that a convex combination of centers in C_i provides a good approximation to $\Delta(O^*_i)$.

The following lemma is a slight modification of the Lemma 2 of [BJK18]. The OPT for cost-balanced k-means is not the sum of 1-means cost of all the optimal clusters. Instead, it is the 1-means cost of the heaviest cluster.

Lemma B.4. $\phi_{C_i}(O^*_i) \leqslant \frac{\varepsilon}{6}\mathsf{OPT}\,.$

Proof. Let $D \stackrel{\text{def}}{=} \sum_{j=1}^k \phi_{C_i}(O^*_j)$. Using the induction hypothesis and the fact that $\phi_{C_i}(O^*_i) \geqslant \phi_{C_i}(O^*_j)$ for $j \geqslant i$, we get that

$$D = \sum_{j=1}^{i-1} \phi_{C_i}(O^*_j) + \sum_{j=i}^{k} \phi_{C_i}(O^*_j) \leqslant \left(1 + \frac{\varepsilon}{2}\right)\sum_{j=1}^{i-1}\Delta(O^*_j) + \frac{\varepsilon k}{2}\mathsf{OPT} + k\phi_{C_i}(O^*_i)\,.$$

Since the Case 1 gives us that $\phi_{C_i}(O^*_i) \leqslant \frac{\varepsilon}{13k}D$, and using the fact that $\Delta(O^*_j) \leqslant \mathsf{OPT}$ for $1 \leqslant j \leqslant k$, we get

$$D \leqslant \frac{\varepsilon}{13}D + (1+\varepsilon)k\mathsf{OPT} \leqslant \left(\frac{(1+\varepsilon)k}{1-\varepsilon/13}\right)\mathsf{OPT}\,.$$

Finally,

$$\phi_{C_i}(O^*_i) \leqslant \frac{\varepsilon}{13k}D \leqslant \frac{\varepsilon}{6}\mathsf{OPT}\,.$$

□

For each point $p \in O^*_i$, let $c(p)$ denote the closest center in C_i. We now define a multi-set O'_i as $\left\{c(p) : p \in O^*_i\right\}$. Note that O'_i is obtained by taking multiple copies of points in C_i. The remaining part of the proof proceeds in two steps. Let m^* and m' denote the denote mean of O^*_i and O'_i respectively. First we will show that m' and m^* are close. Second, we will show that if we have a good approximation m'' to m', then assigning all the points of O^*_i to m'' will incur small cost compared to $\Delta(O^*_i)$. Observe that

$$\sum_{p\in O^*_i} \|p - c(p)\|^2 = \phi_{C_i}(O^*_i)\,.$$

The following lemma is identical to the lemma 3 of [BJK18]

Lemma B.5 (Lemma 3, [BJK18])**.** $\|m^* - m'\|^2 \leqslant \frac{\phi_{C_i}(O^*_i)}{\left|O^*_i\right|}$.

Proof. Let s denote $\left|O^*_i\right|$. Then,

$$\left\|m^* - m'\right\|^2 = \frac{1}{s^2}\left\|\sum_{p\in O^*_i}(p - c(p))\right\|^2 \leqslant \frac{1}{s}\sum_{p\in O^*_i}\|p - c(p)\|^2 = \frac{\phi_{C_i}(O^*_i)}{s}\,,$$

where the inequality follows from triangle inequality followed by Cauchy-Schwarz inequality. □

Next we show that $\Delta(O_i^*)$ and $\Delta(O_i')$ are close (identical to the lemma 4 of [BJK18]).

Lemma B.6 (Lemma 4, [BJK18]). $\Delta(O_i') \leqslant 2\phi_{C_i}(O_i^*) + 2\Delta(O_i^*)$.

Proof. The lemma follows from the following inequalities

$$\begin{aligned}\Delta(O_i') &= \sum_{p\in O_i^*} \left\|c(p) - m'\right\|^2 \leqslant \sum_{p\in O_i^*} \|c(p) - m^*\|^2 \\ &\leqslant 2\sum_{p\in O_i^*} \left(\|c(p) - p\|^2 + \|p - m^*\|^2\right) = 2\phi_{C_i}(O_i^*) + 2\Delta(O_i^*)\,,\end{aligned}$$

where the first inequality follows from Fact B.1 second inequality follows from approximate triangle inequality for squared norm Fact B.2. □

Finally we argue that a good center for O_i' will also serve as a good center for O_i^*.

The following lemma is a slight modification of lemma 5 of [BJK18], where we use the Lemma B.4 instead of Lemma 2 of [BJK18].

Lemma B.7. *Let m'' be a point such that $\phi_{m''}(O_i') \leqslant \left(1 + \frac{\varepsilon}{8}\right)\Delta(O_i')$. Then $\phi_{m''}(O_i^*) \leqslant \left(1 + \frac{\varepsilon}{2}\right)\Delta(O_i^*) + \frac{\varepsilon}{2}\mathsf{OPT}$.*

Proof. Let s^* denote $\left|O_i^*\right|$. Observe that

$$\phi_{m''}(O_i^*) = \sum_{p\in O_i^*} \left\|m'' - p\right\|^2 = \sum_{p\in O_i^*} \|m^* - p\|^2 + s^* \left\|m^* - m''\right\|^2 \qquad \text{(Fact B.1)}\ .$$

$$\begin{aligned}
&\leqslant \Delta(O_i^*) + 2s^*\left(\left\|m^* - m'\right\|^2 + \left\|m' - m''\right\|^2\right) && \text{(Fact B.2)}\\
&\leqslant \Delta(O_i^*) + 2\phi_{C_i}(O_i^*) + 2s^*\left\|m' - m''\right\|^2 && \text{(Lemma B.5)}\\
&\leqslant \Delta(O_i^*) + 2\phi_{C_i}(O_i^*) + 2\left(\phi_{m''}(O_i') - \Delta(O_i')\right) && \text{(Fact B.1)}\\
&\leqslant \Delta(O_i^*) + 2\phi_{C_i}(O_i^*) + \frac{\varepsilon}{4}\Delta(O_i') && \left(\phi_{m''}(O_i') \leqslant \left(1 + \frac{\varepsilon}{8}\right)\Delta(O_i')\right)\\
&\leqslant \Delta(O_i^*) + 2\phi_{C_i}(O_i^*) + \frac{\varepsilon}{2}\left(\phi_{C_i}(O_i^*) + \Delta(O_i^*)\right) && \text{(Lemma B.6)}\\
&\leqslant \left(1 + \frac{\varepsilon}{2}\right)\Delta(O_i^*) + \frac{\varepsilon}{2}\mathsf{OPT}\,. && \text{(Lemma B.4)}
\end{aligned}$$

□

The above lemma tells us that it will be sufficient to obtain a $(1 + \varepsilon/8)$-approximation to the 1-means problem for the data-set O_i'. Lemma B.3 tells us that there is a subset (as a multi-set) O'' of size $\frac{16}{\varepsilon}$ of O_i' such that the mean m'' of these points satisfy the condition of Lemma B.7. Observe that O'' will be a subset of the set S constructed in the step 3 of Algorithm 2. In step 3(c), we add more than $\frac{16}{\varepsilon}$ copies of each point in C_i to S. In step 3(d), the algorithm goes over all subsets of size $\frac{16}{\varepsilon}$ of S, and for each such subset, it tries adding its mean to C_i. In particular, there will be recursive call of this function, where the algorithm will have $C_{i+1} = C_i \cup \{m''\}$ as the set of centers. Lemma B.7 implies that C_{i+1} will satisfy the invariant $P(i + 1)$. Thus, we are done in this case.

Rest of the analysis is identical to [BJK18], as the case 2 (roughly) corresponds to the fact that with good enough probability, we can sample sufficient points from the desired cluster.

Case 2 $\left(\frac{\phi_{C_i}(O_i^*)}{\sum_{j=1}^k \phi_{C_i}(O_j^*)} \geqslant \frac{\varepsilon}{13k}\right)$**:**

We divide the points in O_i^* into two parts: points which are close to a center in C_i, and the remaining points. More formally, let radius R be given by

$$R^2 \stackrel{\text{def}}{=} \frac{\varepsilon^2}{41}\frac{\phi_{C_i}(O_i^*)}{\left|O_i^*\right|}\,. \tag{7}$$

Define O_i^n as the points in O_i^* which are within distance R of a center C_i,

$$O_i^n \stackrel{\text{def}}{=} \left\{ p \in O_i^* : \min_{c \in C_i} \|p - c\| \leqslant R \right\},$$

and O_i^f be the rest of the points in O_i^*, $O_i^f \stackrel{\text{def}}{=} O_i^* \setminus O_i^n$. As in **Case 1**, we define a new set O_i' where each point in O_i^n is replaced by a copy of corresponding point in C_i, ie., for a point $p \in O_i^n$, define $c(p)$ as the closest center in C_i to p. Now define a multi-set O_i' as $O_i^f \cup \left\{c(p) : p \in O_i^n\right\}$. Clearly, $\left|O_i'\right| = \left|O_i^*\right|$. We will argue, that any center that provides a good 1-means approximation for O_i', also provides a good approximation for O_i^*.

Let m^* and m' denote the mean of O_i^* and O_i' respectively. Let s^* and s denote the size of the sets O_i^* and O_i^n respectively. First, we show that $\Delta(O_i^*)$ is large with respect to R.

Lemma B.8 (Lemma 6, [BJK18]). $\Delta(O_i^*) = \phi_{m^*}(O_i^*) \geqslant \frac{16s}{\varepsilon^2} R^2$.

Proof. Let c be the center in C_i which is closest to m^*. We divide the proof into two cases:

(i) $\|m^* - c\| \geqslant \frac{5}{\varepsilon} R$: For any point $p \in O_i^n$, triangle inequality implies that

$$\|p - m^*\| \geqslant \|c(p) - m^*\| - \|c(p) - p\| \geqslant \frac{5}{\varepsilon} R - R \geqslant \frac{4}{\varepsilon} R .$$

Therefore,

$$\Delta(O_i^*) \geqslant \sum_{p \in O_i^n} \|p - m^*\|^2 \geqslant \frac{16s}{\varepsilon^2} R^2 .$$

(ii) $\|m^* - c\| < \frac{5}{\varepsilon} R$: In this case, we have

$$\begin{aligned}
\phi_{m^*}(O_i^*) &= \phi_c(O_i^*) - s^* \|m^* - c\|^2 && \text{(Fact B.1)} \\
&\geqslant \phi_{C_i}(O_i^*) - s^* \|m^* - c\|^2 \\
&\geqslant \frac{41 s^*}{\varepsilon^2} R^2 - \frac{25 s^*}{\varepsilon^2} R^2 && \text{(Using (7))} \\
&\geqslant \frac{16s}{\varepsilon^2} R^2 .
\end{aligned}$$

□

Lemma B.9 (Lemma 7, [BJK18]). $\|m^* - m'\|^2 \leqslant \frac{s}{s^*} R^2$.

Proof. Since the only difference between O_i^* and O_i' is O_i^n, we get

$$\left\|m^* - m'\right\|^2 = \frac{1}{(s^*)^2} \left\| \sum_{p \in O_i^n} (p - c(p)) \right\|^2 \leqslant \frac{s}{(s^*)^2} \sum_{p \in O_i^n} \|p - c(p)\|^2 \leqslant \frac{s^2}{(s^*)^2} R^2 \leqslant \frac{s}{s^*} R^2 ,$$

where the first inequality follows from the Cauchy-Schwartz inequality. □

We now show that $\Delta(O_i')$ is close to $\Delta(O_i^*)$

Lemma B.10 (Lemma 8, [BJK18]). $\Delta(O_i') \leqslant 4sR^2 + 2\Delta(O_i^*)$.

Proof. The lemma follows from the following inequalities.

$$\begin{aligned}
\Delta(O'_{\bar{i}}) &= \sum_{p\in O^n_{\bar{i}}} \left\|c(p)-m'\right\|^2 + \sum_{p\in O^f_{\bar{i}}} \left\|p-m'\right\|^2 \\
&\leqslant \sum_{p\in O^n_{\bar{i}}} 2\left(\|c(p)-p\|^2 + \left\|p-m'\right\|^2\right) + \sum_{p\in O^f_{\bar{i}}} \left\|p-m'\right\|^2 && \text{(Fact B.2)} \\
&\leqslant 2sR^2 + 2\sum_{p\in O^*_{\bar{i}}} \left\|p-m'\right\|^2 \\
&= 2sR^2 + 2\phi_{m'}(O^*_{\bar{i}}) \\
&= 2sR^2 + 2\left(\Delta(O^*_{\bar{i}}) + s^*\left\|m'-m^*\right\|\right) && \text{(Fact B.1)} \\
&\leqslant 4sR^2 + 2\Delta(O^*_{\bar{i}})\,. && \text{(Using Lemma B.9)}
\end{aligned}$$

□

We now argue that any center that is good for $O'_{\bar{i}}$ is also a good center for $O^*_{\bar{i}}$.

Lemma B.11 (Lemma 9, [BJK18]). *Let m'' be such that $\phi_{m''}(O'_{\bar{i}}) \leqslant \left(1+\frac{\varepsilon}{16}\right)\Delta(O'_{\bar{i}})$. Then $\phi_{m''}(O^*_{\bar{i}}) \leqslant \left(1+\frac{\varepsilon}{2}\right)\Delta(O^*_{\bar{i}})$.*

Proof. The lemma follows from the following inequalities.

$$\phi_{m''}(O^*_{\bar{i}}) = \sum_{p\in O^*_{\bar{i}}} \left\|m''-p\right\|^2 = \sum_{p\in O^*_{\bar{i}}} \|m^*-p\|^2 + s^*\left\|m^*-m''\right\|^2 \qquad \text{(Fact B.1)}$$

$$\begin{aligned}
&\leqslant \Delta(O^*_{\bar{i}}) + 2s^*\left(\|m^*-m\|^2 + \left\|m'-m''\right\|^2\right) && \text{(Fact B.2)} \\
&\leqslant \Delta(O^*_{\bar{i}}) + 2s^*\left\|m'-m''\right\|^2 + 2sR^2 && \text{(Lemma B.9)} \\
&\leqslant \Delta(O^*_{\bar{i}}) + 2sR^2 + 2\left(\phi_{m''}(O'_{\bar{i}}) - \Delta(O'_{\bar{i}})\right) && \text{(Fact B.1)} \\
&\leqslant \Delta(O^*_{\bar{i}}) + 2sR^2 + \frac{\varepsilon}{8}\Delta(O'_{\bar{i}}) \\
&\leqslant \Delta(O^*_{\bar{i}}) + 2sR^2 + \frac{\varepsilon}{2}sR^2 + \frac{\varepsilon}{4}\Delta(O^*_{\bar{i}}) && \text{(Lemma B.10)} \\
&\leqslant \left(1+\frac{\varepsilon}{2}\right)\Delta(O^*_{\bar{i}})\,. && \text{(Lemma B.8)}
\end{aligned}$$

□

Given the above lemma, all we need to argue is that our algorithm considers a center m'' such that $\phi_{m''}(O'_{\bar{i}}) \leqslant \left(1+\frac{\varepsilon}{16}\right)\Delta(O'_{\bar{i}})$. For this we would need about $O(1/\varepsilon)$ uniform samples from $O'_{\bar{i}}$. However, our algorithm can only sample only using D^2-sampling w.r.t. C_i. For ease of notation, let $c(O^n_{\bar{i}})$ denote the multiset $\left\{c(p) : p \in O^n_{\bar{i}}\right\}$. Recall that $O'_{\bar{i}}$ consists of $O^f_{\bar{i}}$ and $c(O^n_{\bar{i}})$. The first observation is that the probability of sampling an element from $O^f_{\bar{i}}$ is reasonably large (proportional to ε/k). Using this fact, it can be shown how to sample from $O'_{\bar{i}}$ almost uniformly. The second observation is that one can convert the previous almost uniform sampling to uniform sampling, (at the cost of increasing the size of the sample). The rest of the details of the sampling follows from [BJK18]. We include those for completeness sake.

Lemma B.12 (Lemma 10, [BJK18]). *Let x be a sample from D^2-sampling w.r.t. C_i. Then, $\mathbb{P}\left[x \in O^f_{\bar{i}}\right] \geqslant \frac{\varepsilon}{15k}$. Further for any point $p \in O^f_{\bar{i}}$, $\mathbb{P}[x=p] \geqslant \frac{\gamma}{\left|O^*_{\bar{i}}\right|}$, where $\gamma = \frac{\varepsilon^2}{533k}$.*

Proof. Note that $\sum_{p \in O_i^* \setminus O_i^f} \mathbb{P}[x = p] \leqslant \frac{R^2}{\phi_{C_i}(X)} \left|O_i^*\right| \leqslant \frac{\varepsilon^2}{41} \frac{\phi_{C_i}(O_i^*)}{\phi_{C_i}(X)}$. Therefore, the fact that we are in the **Case 2** implies that

$$\mathbb{P}\left[x \in O_i^f\right] \geqslant \mathbb{P}[x \in O_i^*] - \mathbb{P}\left[x \in O_i^* \setminus O_i^f\right] \geqslant \frac{\phi_{C_i}(O_i^*)}{\phi_{C_i}(X)} - \frac{\varepsilon^2}{41} \frac{\phi_{C_i}(O_i^*)}{\phi_{C_i}(X)} \geqslant \frac{\varepsilon}{15k}\,.$$

Also if $x \in O_i^f$, then $\phi_{C_i}(\{x\}) \geqslant R^2 = \frac{\varepsilon^2}{41} \frac{\phi_{C_i}(O_i^*)}{\left|O_i^*\right|}$. Therefore,

$$\frac{\phi_{C_i}(\{x\})}{\phi_{C_i}(X)} \geqslant \frac{\varepsilon}{13k} \frac{R^2}{\phi_{C_i}(O_i^*)} \geqslant \frac{\varepsilon}{13k} \frac{\varepsilon^2}{41} \frac{1}{\left|O_i^*\right|} \geqslant \frac{\varepsilon^2}{533k} \frac{1}{\left|O_i^*\right|}\,.$$

This completes the proof of the lemma. □

Let $X_1, \ldots, X_l$ be l points sampled independently using D^2-sampling w.r.t. C_i. We construct a new set of random variables $Y_1, \ldots, Y_l$. Each variable Y_u will depend only on X_u, and will take values either in O_i' or will be $\perp$. These variable are defined as follows: if $X_u \notin O_i^f$, we set Y_u to $\perp$, otherwise, we assign Y_u to one of the following random variables with equal probability: (i) X_u or (ii) a uniformly at random element of the multi-set $c(O_i^n)$. The following observation follows from the Lemma B.12.

Corollary B.13 (Corollary 2, [BJK18]). *For a fixed index u, and an element $x \in O_i'$, $\mathbb{P}[Y_u = x] \geqslant \frac{\gamma'}{\left|O_i'\right|}$, where $\gamma' = \gamma/2$.*

Proof. If $x \in O_i^f$, them we know from Lemma B.12 that X_u is x with probability at least $\frac{\gamma}{\left|O_i'\right|}$. Conditioned on this event, $Y_u = X_u$ with probability $1/2$. Now suppose $x \in c(O_i^n)$, Lemma B.12 implies that X_u is an element of O_i^f with probability at least $\frac{\varepsilon}{15k}$. Conditioned on this event, Y_u will be equal to x with probability at least $\frac{1}{2} \frac{1}{\left|c(O_i^n)\right|}$. Therefore

$$\mathbb{P}[X_u = x] \geqslant \frac{\varepsilon}{15k} \frac{1}{2\left|c(O_i^n)\right|} \geqslant \frac{\varepsilon}{30k\left|O_i'\right|} \geqslant \frac{\gamma'}{\left|O_i'\right|}\,.$$

□

Corollary B.13 shows that we can obtain samples from O_i' which are nearly uniform (up to a constant factor). To convert this to a set of uniform samples, we use the idea of [JKS14]. For an element $x \in O_i'$, let γ_x be such that $\frac{\gamma_x}{\left|O_i'\right|}$ denotes the probability that the random variable Y_u is equal to x. Corollary B.13 implies that $\gamma_x \geqslant \gamma'$. We define a new set of independent random variables $Z_1, \ldots, Z_l$. The random variable Z_u will depend only on Y_u. If Y_u is $\perp$ then Z_u is $\perp$. If Y_u is equal to $x \in O_i'$, then Z_u take the value x with probability $\frac{\gamma'}{\gamma_x}$, and $\perp$ with remaining probability. Note that Z_u is either $\perp$ or one of the elements of O_i'. Further, conditioned on the latter event, it is a uniform sample from O_i'. We can now prove the key lemma.

Lemma B.14 (Lemma 11, [BJK18]). *Let l be $\frac{128}{\gamma'\varepsilon}$, and m'' denote the mean of non-null samples from $Z_1, \ldots, Z_l$. Then with probability at least $1/2$, $\phi_{m''}(O_i') \leqslant (1 + \varepsilon/16)\Delta(O_i')$.*

Proof. Note that random variable Z_u is equal to a specific element of O_i' with probability equal to $\frac{\gamma'}{\left|O_i'\right|}$, and it takes $\perp$ with probability $1 - \gamma'$. Now consider a different set of iid random variables Z_u', $1 \leqslant u \leqslant l$ as follows: each Z_u tosses a coin with probability of heads being γ'. If we get heads, it gets value $\perp$, otherwise it is equal to a random element of O_i'. The joint distribution of the random variable Z_u' is identical to that of the random variable Z_u. Thus it suffices to prove the statement of the lemma for random variable Z_u'.

Now we condition on coin tosses of the random variables Z_u'. Let s' be the number of random variables which are not $\perp$ (s' is a deterministic quantity because we have conditioned on coin tosses). Let m'' be the mean of such non-$\perp$ variables among $Z_1', \ldots, Z_l'$. If s' happens to be larger than $64/\varepsilon$, Lemma B.3 implies that with probability at least $3/4$, $\phi_{m''}(O_i') \leqslant (1 + \varepsilon/16)\Delta(O_i')$.

Finally, observe that the expected number of non-$\perp$ variables is $\gamma' l \geqslant 128/\varepsilon$. Therefore, with probability at least $3/4$, the number of non-$\perp$ elements will be at least $64/\varepsilon$. □

Let $C_i^{(l)}$ denote the multi-set obtained by taking l copies of each of the centers in C_i. Now observe that all the non-$\perp$ elements among $Y_1, \ldots, Y_l$ are elements of $\{X_1, \ldots, X_l\} \cup C_i^{(l)}$, and so the same must hold for $Z_1, \ldots, Z_l$. This implies that in Step 3(d) of Algorithm 1, we would have tried adding the point m'' as described in Lemma B.14. Therefore, the induction hypothesis continues to hold with probability at least 1/2.

Runtime Analysis: Recall that the size of the list $\mathcal{L}$ constructed by Algorithm 1 is $2^{\tilde{O}(k/\varepsilon)}$, and therefore the running time of the algorithm is $O\left(nd\, 2^{\tilde{O}(k/\varepsilon)}\right)$. □

C $(1+\varepsilon)$-Approximation for Cost-Balanced k-Median Clustering

The setting for the cost-balanced k-median is same as that for cost-balanced k-means problem, except for the fact that the distances are measured using the Euclidean norm. The notations are as same as before, modified for the k-median problem.

Notations: Let $\Delta(X)$ denote the 1-median cost of the set of points, i.e., $\Delta(X) \stackrel{\text{def}}{=} \min_{c \in \mathbb{R}^d} \sum_{x \in X} \|x - c\|$. A k-partition of X into disjoint subsets $\mathbb{O} = \{O_1, \ldots, O_k\}$ is called a k-clustering of X. We denote the optimal cost-balanced k-median clustering by $\mathbb{O}^* = \left\{O_1^*, \ldots, O_k^*\right\}$. Given a clustering $\mathbb{O}$ and a set $C = \{c_1, \ldots, c_k\}$, we define $cost_C(\mathbb{O})$ as the minimum over all permutation π of C of $\max_{i \in [k]} \sum_{x \in O_i} \left\|x - c_{\pi(i)}\right\|$. Let OPT denotes the optimal value of the cost-balanced k-median. For a set of points X and another set of points C, we define $\phi_C(X) = \sum_{x \in X} \min_{c \in C} \|x - c\|$. With a slight abuse of notation, when set C has only one element c, we will use the notation $\phi_c(X)$, instead of $\phi_{\{c\}}(X)$.

For cost-balanced k-median, we no longer have an analogue of the Fact B.1, i.e., for a set of points X, if c^* denotes the optimal center w.r.t. the 1-median problem, and c is a point such that $\phi_c(X) \leqslant (1+\varepsilon)\phi_{c^*}(X)$, it is possible that $\|c - c^*\|$ is large. This in turn implies that there is no analogue of the Lemma B.3. However, instead of the approximate triangle inequality Fact B.2, we get a triangle inequality in the Euclidean metric.

Lemma C.1 (Theorem 5.4, [KSS10])**.** *Given a random sample (with replacement) R of size $1/\varepsilon^4$ from a set of points $X \in \mathbb{R}^d$, there is a procedure construct(R), which outputs a set core(R) of size $2^{(1/\varepsilon)^{O(1)}}$ such that the following event happens with probability at least $1/2$: there is at least one point $c \in$ core(R) such that $\phi_c(X) \leqslant (1+\varepsilon) \cdot \Delta(X)$. The time taken by the procedure construct(R) is $O\left(2^{(1/\varepsilon)^{O(1)}} \cdot d\right)$.*

Here, we will not sample according to the D^2-sampling, but according to the D-sampling defined below.

Definition C.2 (D-sampling)**.** Given a set of points $X \subset \mathbb{R}^d$ and another set of points $C \subset \mathbb{R}^d$, D-sampling from X w.r.t. C samples a point $x \in X$ with probability $\frac{\phi_C(\{x\})}{\phi_C(X)}$. When $C = \emptyset$, we pick a point uniformly at random from X.

The algorithm for the cost-balanced k-median is same as that Algorithm 1 and Algorithm 2 except for some minor changes in both the algorithms. The parameters α and β in Algorithm 4 and Algorithm 5 are large enough constants.

Proof of Theorem 1.7. In this proof, we will mention the changes that are needed from the analysis of the cost-balanced k-means clustering. The core of the analysis remains the same. The proof follows with some minor modifications of the Section 5 of [BJK18]. We give the whole proof for completeness sake.

We would like to prove the induction hypothesis $P(i)$. We use the same notation as Section B, and define **Case 1** and **Case 2** analogously. Consider the **Case 1** first. Proof of the Lemma B.4 remains unchanged. The set $O'_{\bar{i}}$ is define similarly. Let m^* be the point for which $\Delta(O^*_{\bar{i}}) = \phi_{m^*}(O^*_{\bar{i}})$. Define m' analogously for the set $O'_{\bar{i}}$. The statement of the Lemma B.6 changes as follows:

$$\begin{aligned}\Delta(O'_{\bar{i}}) &\leqslant \sum_{p \in O^*_{\bar{i}}} \left\|c(p) - m'\right\| \leqslant \sum_{p \in O^*_{\bar{i}}} \|c(p) - m^*\| \leqslant \sum_{p \in O^*_{\bar{i}}} (\|c(p) - p\| + \|p - m^*\|) \\ &= \phi_{C_i}(O^*_{\bar{i}}) + \Delta(O^*_{\bar{i}}) \end{aligned} \tag{8}$$

Algorithm 4: Cost-Balanced k-Median Algorithm

Input: Set of points $X \subset \mathbb{R}^d$, number of clusters k, and an error parameter ε.
Output: A cost-balanced k means clustering $\mathbb{O}^A$.

1. Let $N = \frac{\alpha\, k}{\varepsilon^6}$, $M = \frac{\beta}{\varepsilon^4}$.
2. Initialize $\mathcal{L}$ to $\emptyset$. $\mathcal{L}$ will contain a list of candidate means of a clustering, where each candidate mean is a set of *exactly* k centers.
3. Repeat 2^k times:
   - Make a call to (Algorithm 5) Sample-Centers$(X, k, \varepsilon, \mathcal{L}, 0, \{\})$.
4. For each tuple t in $\mathcal{L}$:
   - Form a matrix $\mathcal{J}_{[k \times n]}$, where $\mathcal{J}(i, j) = \left\| t_i - x_j \right\|$.
   - Input $t, \mathcal{J}$ to the Jansen & Mastrolilli's [JM10] algorithm for minimum makespan scheduling on unrelated machines.
   - Maintain the clustering with the minimum cost.
5. Return the minimum cost clustering $\mathbb{O}^A$.

Algorithm 5: Sample-Centers Algorithm (Subroutine of Algorithm 5.1, [BJK18])

Input: Set of points $X \subset \mathbb{R}^d$, number of clusters k, an error parameter ε, a list $\mathcal{L}$ of k-tuples, index i , and a set C of centers.

1. Set $N = \frac{\alpha\, k}{\varepsilon^6}$, $M = \frac{\beta}{\varepsilon^4}$, $S' = \emptyset$.
2. If $(i = k)$ then add C to the set $\mathcal{L}$.
3. else
   (a) S is an i.i.d. sample of N points picked by D-sampling (Definition C.2) w.r.t. C.
   (b) $S' \leftarrow S$.
   (c) For all $c \in C : S' \leftarrow S' \cup \{M \text{ copies of } c\}$.
   (d) For all subsets T which is a collection of M points from S' (with repetitions allowed) and for all elements $c \in \text{core}(T)$:
      i. $C \leftarrow C \cup \{c\}$.
      ii. Sample-Centers$(X, k, \varepsilon, \mathcal{L}, i + 1, C)$.

Proof of Lemma B.7 also changes as follows: let m'' be as in the statement of this lemma. Then,

$$\begin{aligned}
\phi_{m''}(O_i^*) &= \sum_{p\in O_i^*} \left\|p - m''\right\| \\
&\leqslant \sum_{p\in O_i^*} \left(\|p - c(p)\| + \left\|c(p) - m''\right\|\right) \\
&= \phi_{C_i}(O_i^*) + \phi_{m''}(O_i') \\
&\leqslant \phi_{C_i}(O_i^*) + \left(1 + \frac{\varepsilon}{8}\right)\Delta(O_i') \\
&\leqslant 2\phi_{C_i}(O_i^*) + \left(1 + \frac{\varepsilon}{8}\right)\Delta(O_i^*) && \text{(Using (8))} \\
&\leqslant \frac{\varepsilon}{3}\mathsf{OPT} + \left(1 + \frac{\varepsilon}{8}\right)\Delta(O_i^*) && \text{(Using Lemma B.4)}
\end{aligned}$$

Rest of the arguments remain unchanged (we use Lemma C.1 instead of Lemma B.3). Now we consider the **Case 2**. We redefine the parameter R as

$$R \stackrel{\text{def}}{=} \frac{\varepsilon}{9} \cdot \frac{\phi_{C_i}(O_i^*)}{\left|O_i^*\right|}.$$

Define sets $O_i', c(O_i^n), O_i^f$ as before. Let m^* be the point for which $\Delta(O_i^*) = \phi_{m^*}(O_i^*)$, and m' be the analogous point for O_i'. Proof of Lemma B.8 can be easily modified to yield the following (we just need to use the triangle inequality instead of the Fact B.1):

$$\Delta(O_i^*) = \phi_{m^*}(O_i^*) \geqslant \frac{4s}{\varepsilon}R. \tag{9}$$

We have the following version of the Lemma B.10:

$$\begin{aligned}
\Delta(O_i') \leqslant \phi_{m^*}(O_i') &= \sum_{p\in O_i^n} \|c(p) - m^*\| + \sum_{p\in O_i^f} \|p - m^*\| \\
&\leqslant \sum_{p\in O_i^n} (\|p - m^*\| + \|c(p) - p\|) + \sum_{p\in O_i^f} \|p - m^*\| \\
&\leqslant sR + \Delta(O_i^n), && (10)
\end{aligned}$$

where s denotes $\left|O_i^n\right|$. Finally, let m'' be as in the statement of the Lemma B.11. Then,

$$\begin{aligned}
\phi_{m''}(O_i^*) &= \sum_{p\in O_i^n} \left\|p - m''\right\| + \sum_{p\in O_i^f} \left\|p - m''\right\| \\
&\leqslant \sum_{p\in O_i^n} \left(\left\|c(p) - m''\right\| + \|c(p) - p\|\right) + \sum_{p\in O_i^f} \left\|p - m''\right\| \\
&\leqslant sR + \phi_{m''}(O_i') \leqslant sR + \left(1 + \frac{\varepsilon}{8}\right)\Delta(O_i') \\
&\leqslant 3sR + \left(1 + \frac{\varepsilon}{8}\right)\Delta(O_i^*) && \text{(Using (10))} \\
&\leqslant (1 + \varepsilon)\Delta(O_i^*). && \text{(Using (9))}
\end{aligned}$$

Rest of the arguments go through without any changes. □

References

[ABF+18] Sara Ahmadian, Babak Behsaz, Zachary Friggstad, Amin Jorati, Mohammad R. Salavatipour, and Chaitanya Swamy, *Approximation algorithms for minimum-load k-facility location*, ACM Trans. Algorithms **14** (2018), no. 2, 16:1–16:29. 4

[ABS10] Marcel R. Ackermann, Johannes Blömer, and Christian Sohler, *Clustering for metric and nonmetric distance measures*, ACM Trans. Algorithms **6** (2010), no. 4, 59:1–59:26. 4

[AHL06] Esther M. Arkin, Refael Hassin, and Asaf Levin, *Approximations for minimum and min-max vehicle routing problems*, Journal of Algorithms **59** (2006), no. 1, 1 – 18. 4

[ANFSW17] S. Ahmadian, A. Norouzi-Fard, O. Svensson, and J. Ward, *Better guarantees for k-means and euclidean k-median by primal-dual algorithms*, 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), Oct 2017, pp. 61–72. 3

[AV07] David Arthur and Sergei Vassilvitskii, *K-means++: The advantages of careful seeding*, Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. 3

[BJK18] Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar, *Faster algorithms for the constrained k-means problem*, Theory of Computing Systems **62** (2018), no. 1, 93–115. 4, 5, 6, 11, 12, 13, 14, 15, 16, 17, 18

[BPR+17] Jaroslaw Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh, *An improved approximation for k-median and positive correlation in budgeted optimization*, ACM Trans. Algorithms **13** (2017), no. 2, 23:1–23:31. 3

[CAKM16] V. Cohen-Addad, P. N. Klein, and C. Mathieu, *Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics*, 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), Oct 2016, pp. 353–364. 3

[CGTS02] Moses Charikar, Sudipto Guha, va Tardos, and David B. Shmoys, *A constant-factor approximation algorithm for the k-median problem*, Journal of Computer and System Sciences **65** (2002), no. 1, 129 – 149. 3

[Cha18] Moses Charikar, *Personal communication*. 5, 10

[Che09] Ke Chen, *On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications*, SIAM J. Comput. **39** (2009), no. 3, 923–947. 3, 4, 5

[CS19] Deeparnab Chakrabarty and Chaitanya Swamy, *Approximation algorithms for minimum norm and ordered optimization problems*, Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (New York, NY, USA), STOC 2019, ACM, 2019, pp. 126–137. 3

[DLS19] Amit Deshpande, Anand Louis, and Apoorv Singh, *On euclidean k-means clustering with alpha-center proximity*, Proceedings of Machine Learning Research (Kamalika Chaudhuri and Masashi Sugiyama, eds.), Proceedings of Machine Learning Research, vol. 89, PMLR, 16–18 Apr 2019, pp. 2087–2095. 4

[DX15] Hu Ding and Jinhui Xu, *A unified framework for clustering constrained data without locality property*, Proceedings of the Twenty-sixth Annual ACM-SIAM Symposium on Discrete Algorithms (Philadelphia, PA, USA), SODA '15, Society for Industrial and Applied Mathematics, 2015, pp. 1471–1490. 4

[EGK+03] G. Even, N. Garg, J. Könemann, R. Ravi, and A. Sinha, *Covering graphs using trees and stars*, Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques (Berlin, Heidelberg) (Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai, eds.), Springer Berlin Heidelberg, 2003, pp. 24–35. 4

[FJM08] Aleksei V. Fishkin, Klaus Jansen, and Monaldo Mastrolilli, *Grouping techniques for scheduling problems: Simpler and faster*, Algorithmica **51** (2008), no. 2, 183–199. 4

[FMS07] Dan Feldman, Morteza Monemizadeh, and Christian Sohler, *A ptas for k-means clustering based on weak coresets*, Proceedings of the Twenty-third Annual Symposium on Computational Geometry, SCG '07, ACM, 2007, pp. 11–18. 3, 4, 5

[FRS16] Z. Friggstad, M. Rezapour, and M. R. Salavatipour, *Local search yields a ptas for k-means in doubling metrics*, 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), Oct 2016, pp. 365–374. 3

[GJ90] Michael R. Garey and David S. Johnson, *Computers and intractability; a guide to the theory of np-completeness*, W. H. Freeman & Co., New York, NY, USA, 1990. 4, 10

[HPK05] Sariel Har-Peled and Akash Kushal, *Smaller coresets for k-median and k-means clustering*, Proceedings of the Twenty-first Annual Symposium on Computational Geometry, SCG '05, ACM, 2005, pp. 126–134. 3, 5

[HPM04] Sariel Har-Peled and Soham Mazumdar, *On coresets for k-means and k-median clustering*, Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing, STOC '04, ACM, 2004, pp. 291–300. 3, 5

[HS76] Ellis Horowitz and Sartaj Sahni, *Exact and approximate algorithms for scheduling nonidentical processors*, J. ACM **23** (1976), no. 2, 317–327. 4

[HS87] Dorit S. Hochbaum and David B. Shmoys, *Using dual approximation algorithms for scheduling problems theoretical and practical results*, J. ACM **34** (1987), no. 1, 144–162. 4

[IKI94] Mary Inaba, Naoki Katoh, and Hiroshi Imai, *Applications of weighted voronoi diagrams and randomization to variance-based k-clustering: (extended abstract)*, Proceedings of the Tenth Annual Symposium on Computational Geometry (New York, NY, USA), SCG '94, ACM, 1994, pp. 332–339. 3, 11

[JKS14] Ragesh Jaiswal, Amit Kumar, and Sandeep Sen, *A simple d 2-sampling based ptas for k-means and other clustering problems*, Algorithmica **70** (2014), no. 1, 22–46. 4, 16

[JM10] Klaus Jansen and Monaldo Mastrolilli, *Scheduling unrelated parallel machines: linear programming strikes back*, Technische Berichte des Instituts für Informatik der CAU Kiel **TR_1004** (2010). 4, 5, 6, 7, 8, 18

[JP01] Klaus Jansen and Lorant Porkolab, *Improved approximation schemes for scheduling unrelated parallel machines*, Mathematics of Operations Research **26** (2001), no. 2, 324–338. 4

[JV01] Kamal Jain and Vijay V. Vazirani, *Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation*, J. ACM **48** (2001), no. 2, 274–296. 3

[KMN+04] 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*, Computational Geometry **28** (2004), no. 2, 89 – 112, Special Issue on the 18th Annual Symposium on Computational Geometry - SoCG2002. 3

[KSS04] Amit Kumar, Yogish Sabharwal, and Sandeep Sen, *A simple linear time (1+ε)-approximation algorithm for k-means clustering in any dimensions*, 45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings, 2004, pp. 454–462. 3, 4, 5

[KSS10] Amit Kumar, Yogish Sabharwal, and Sandeep Sen, *Linear-time approximation schemes for clustering problems in any dimensions*, J. ACM **57** (2010), no. 2, 5:1–5:32. 5, 17

[L.96] Graham R. L., *Bounds for certain multiprocessing anomalies*, Bell System Technical Journal **45** (1996), no. 9, 1563–1581. 4

[LS13] Shi Li and Ola Svensson, *Approximating k-median via pseudo-approximation*, Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing (New York, NY, USA), STOC '13, ACM, 2013, pp. 901–910. 3

[LST90] Jan Karel Lenstra, David B. Shmoys, and Éva Tardos, *Approximation algorithms for scheduling unrelated parallel machines*, Mathematical Programming **46** (1990), no. 1, 259–271. 4

[Vaz03] Vijay V. Vazirani, *Approximation algorithms*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. 9

[VKKR03] W. Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani, *Approximation schemes for clustering problems*, Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing, STOC '03, ACM, 2003, pp. 50–58. 3, 5