Tight FPT Approximations for k-Median and k-Means

Vincent Cohen-Addad
CNRS & Sorbonne Université„ Paris, France

Anupam Gupta
Carnegie Mellon University, Pittsburgh, PA, USA

Amit Kumar
IIT Delhi, India

Euiwoong Lee
New York University, NY, USA

Jason Li
Carnegie Mellon University, Pittsburgh, PA, USA

──── **Abstract** ────

We investigate the fine-grained complexity of approximating the classical k-MEDIAN/k-MEANS clustering problems in general metric spaces. We show how to improve the approximation factors to $(1 + 2/e + \varepsilon)$ and $(1 + 8/e + \varepsilon)$ respectively, using algorithms that run in fixed-parameter time. Moreover, we show that we cannot do better in FPT time, modulo recent complexity-theoretic conjectures.

1 Introduction

How well can we approximate the k-MEDIAN and k-MEANS clustering problems? This question has been intensively studied over the past two decades, and many interesting algorithmic techniques have been developed and refined in an attempt to understand these problems. Let us elaborate for the k-MEDIAN problem; the story for k-MEANS is much the

same. Recall that in the k-MEDIAN problem, given a metric space (V, d) with n points and clients at some of the points, the goal is to open k *facilities* such that the sum of distances from the clients to their closest facilities is minimized.

The first constant-factor approximation algorithm for k-MEDIAN was given by Charikar et al. [6]. After many interesting developments (e.g., primal-dual schemes, sophisticated LP rounding schemes, and pseudo-approximations), today the best approximation guarantee is 2.611 [3]. The best lower bound, however, is still the $(1 + 2/e)$-hardness from 1998, due to Guha and Khuller [16]. In this paper, we ask: can we do better if we give ourselves more resources? The problem can be solved exactly by brute-force enumeration in time $n^{k+O(1)}$, but what can we do, say, in FPT time $f(k)n^{O(1)}$?

We cannot hope to solve the problem exactly in FPT time: the reduction of Guha and Khuller also shows a $W[2]$-hardness for finding the optimal solution for k-MEDIAN/k-MEANS exactly. Naturally, we then ask what we can achieve by combining the two approaches together, and whether good approximation algorithms can be given in FPT time.

Our Results.   Our main algorithmic result is a positive result in this direction:

▶ **Theorem 1** (Algorithm for k-MEDIAN/k-MEANS)**.** *For every $\varepsilon > 0$, there is a $(1+2/e+\varepsilon)$-approximation algorithm for the k-MEDIAN problem, that runs in time FPT time, i.e., in $f(k, \varepsilon)n^{O(1)}$ time. For the k-MEANS problem, we can achieve a $(1 + 8/e + \varepsilon)$-approximation in the same runtime.*

The approximation guarantees in Theorem 1 match the NP-hardness results for the two problems implied by [16]. However, since we are allowing ourselves FPT time and not just $\text{poly}(n, k)$ time, can we do even better and go past this NP-hardness barrier? Our second main result shows that this is not possible, at least under recent complexity-theoretic conjectures. We prove that the results in Theorem 1 are essentially tight, assuming the Gap-Exponential Time Hypothesis [12, 24, 5]:

▶ **Theorem 2** (Hardness)**.** *There exists a function $g : \mathbb{R}^+ \to \mathbb{R}^+$ such that assuming the Gap-ETH, for any $\varepsilon > 0$, any $(1 + 2/e - \varepsilon)$-approximation algorithm for k-MEDIAN, and any $(1 + 8/e - \varepsilon)$-approximation for k-MEANS, must run in time at least $n^{k^{g(\varepsilon)}}$.*

The basic component of the above hardness result is an FPT-hardness of a factor of $(1 - 1/e)$ for the MAX k-COVERAGE problem, again using the Gap-ETH (Theorem 15). Composing that hardness result with the reduction of Guha and Khuller [16] gives us Theorem 2 above.

Matroid Median.   Finally, using our algorithmic techniques, we are able to also give an improved approximation for the matroid-median problem, which is a generalization of the k-MEDIAN problem.

▶ **Theorem 3** (Algorithm for MATROID MEDIAN)**.** *There is a $(2+\varepsilon)$-approximation algorithm for the MATROID MEDIAN problem, that runs in time FPT time, i.e., in $f(k, \varepsilon)n^{O(1)}$ time.*

Since the MATROID MEDIAN problem is a generalization of the k-MEDIAN problem, the $(1 + 2/e - \varepsilon)$-hardness from Theorem 2 translates immediately to MATROID MEDIAN. It remains an open problem to close the gap between this lower bound and the $(2 + \varepsilon)$-approximation in Theorem 3. We can also use our ideas to get an $(3 - \frac{2}{p+1} + \varepsilon)$ for the p-MATROID MEDIAN problem.

Facility Location. FACILITY LOCATION is a problem closely related to k-MEDIAN, where each facility has an opening cost and the goal is to open facilities to minimize the sum of distances from clients to their closest facilities plus the sum of the total opening costs. For this problem, the best known hardness ratio is $\alpha_{\mathsf{FL}} \approx 1.463$ [16], which is defined to be $\max_{x \geq 0} \left(1 + \frac{x}{1+x} \ln \frac{2}{x}\right)$. On the other hand, the best algorithm achieves an 1.488-approximation [21]. When the parameter k denotes the number of facilities open in the optimal solution, we prove that our techniques also give an FPT algorithm for FACILITY LOCATION whose approximation ratio matches the hardness ratio of [16].

▶ **Theorem 4** (Algorithm for FACILITY LOCATION). *There is a $(\alpha_{\mathsf{FL}} + \varepsilon)$-approximation algorithm for the* FACILITY LOCATION *problem, that runs in time FPT time, i.e., in* $f(k, \varepsilon)n^{O(1)}$ *time.*

Roadmap. In Section 2, we describe the approximation algorithms for these problems. We assume throughout that the aspect ratio is polynomially bounded. (In the full version of the paper, we show that this assumption is without loss of generality, in the case we consider where the clients have unit weights.) In Section 3, we then give the hardness results for FPT MAX k-COVERAGE, k-MEDIAN, and k-MEANS.

1.1 Our Techniques

The algorithm is inspired by the hardness result from [16]: it relies on the result of Feige [13] that MAX k-COVERAGE is hard to approximate better than $(1 - 1/e)$. Hence, if we build a "factor graph" with sets on one side and elements on another, with edges indicating inclusion, picking k sets covers $(1 - 1/e)$ elements at distance 1, and the remaining at distance at least 3 – hence $1 + 2/e$. Now what if we have a general instance, with different distances? We show how to do limited enumeration (in FPT) time to restrict our choices to picking one facility each from k disjoint sets. Moreover, via a surprisingly clean idea we can model the objective as submodular maximization (subject to a partition matroid constraint). And this problem can be approximated well: the factor again is $(1 - 1/e)$, hence giving the same factor upto additive ε terms!

The matching hardness result is via showing an FPT hardness for MAX k-COVERAGE assuming the Gap-ETH. Firstly, we show that assuming the Gap-ETH, there is no FPT approximation algorithm for LABEL COVER problem parameterized by the number of vertices k on one side of the bipartition. (Trying all labelings on one side takes time $O(n^{k+O(1)})$, and doing much better is hard.) To do this, we construct a *variable-clause game* from a 3-SAT instance, merge *clause vertices* into ℓ super-vertices, and then use r rounds of parallel repetition. (The number of clause vertices becomes $k := \ell^r$.) Then we compose this with the classical reduction from LABEL COVER to MAX k-COVERAGE [13]. Due to some technical details (e.g., our LABEL COVER instance is not guaranteed to be regular) and for the sake of completeness, we provide a formal proof in Lemma 19. While our techniques are similar to recent FPT hardnesses for the related k-DOMINATING SET problem [5, 10], some technical details (e.g., the *projection property* of LABEL COVER instances) prevent us from directly using prior results to get $(1 - 1/e + \varepsilon)$-hardness for MAX k-COVERAGE.

1.2 Related Work

We briefly survey the state-of-the-art for k-MEDIAN and k-MEANS; please see references below for more historical context. For general metric spaces, the best approximation ratio for k-MEDIAN is 2.611 [3] by Byrka et al., building on work of Li and Svensson [22]. Kanungo et al. [17] gave a $(9 + \varepsilon)$-approximation algorithm for k-MEANS in general metric spaces, which

was later improved to 6.357 by Ahmadian et al. [1]. The first constant factor approximation algorithm for MATROID MEDIAN was given by Krishnaswamy et al. [18], which was improved by Swamy [26] to 8.

For Euclidean spaces, the problems are better approximable, at least when either k or the dimension d are fixed; we restrict this discussion to parameterizing by k. Specifically, PTASs for both k-MEDIAN and k-MEANS with running time $f(k, \varepsilon) \operatorname{poly}(n, d)$ were given by Kumar et al. [19]. The running times were improved by Chen [7] to $O(nkd + d^2 n^\sigma 2^{(k/\varepsilon)^{O(1)}})$ for any $\sigma > 0$ for k-MEDIAN, and by Feldman et al. [15] to $O(nkd + d\operatorname{poly}(k/\varepsilon) + 2^{\tilde{O}(k/\varepsilon)})$ for k-MEANS. Both these latter results were based on the notion of coresets. The k-MEANS problem is APX hard even in Euclidean space, if both k and d are allowed to be arbitrary [2, 20].

A result of direct interest to this work is that of Czumaj and Sohler [11], for the *min-sum clustering problem*. They give a $(4 + \varepsilon)$-approximation on general metrics in FPT time. They construct a small (strong) core-set for the related BALANCED k-MEDIAN problem, and enumerate over all choices of centers inside this core-set. In the full version, we show that their approach extends to give a 2-approximation for the *non-bipartite case* of k-MEDIAN– in this special case of k-MEDIAN a facility may be opened at any client location, and hence $C \subseteq \mathbb{F}$. Theorem 1 above shows how to get a better guarantee for a more general case. (As an aside, the hardness for this special non-bipartite case is only $(1 + 1/e)$; closing this gap is another interesting open question.)

Hardness-of-approximation results for parameterized problems have been actively studied recently. Lin [23] proved $W[1]$-hardness of approximation for k-BICLIQUE. Chen and Lin [9] proved $W[1]$-hardness of approximation for k-DOMINATING SET in any constant factor, which was later improved to any function $f(k)$ in [5, 10]. Chalermsook et al. [5] also proved that there is no FPT $o(k)$-approximation algorithm for k-CLIQUE assuming the Gap-ETH.

1.3   Preliminaries

An instance $\mathcal{I}$ of the k-MEDIAN problem is defined by a tuple $((V, d), C, \mathbb{F}, k)$, where (V, d) is a metric space over a set of points V with $d(i, j)$ denoting the distance between two points i, j in V. Further, C and $\mathbb{F}$ are subsets of V and are referred as "clients" and "facility locations", and k is a positive parameter. The goal is to find a subset F of k facilities in $\mathbb{F}$ to minimize

$$\operatorname{cost}(C, F) := \sum_{j \in C} d(j, F).$$

In the *weighted* version of k-MEDIAN, every client $j \in C$ has an associated weight w_j, and the goal is to find a subset F of $\mathbb{F}$ of size k such that $\operatorname{cost}(C, F) := \sum_{j \in C} w_j d(j, F)$ is minimized.

The k-MEANS problem is defined similarly except that the objective function gets modified to $\operatorname{cost}(C, F) := \sum_{j \in C} d(j, F)^2$ (and analogously for the weighted version). The names of the two problems come from the fact that if the metric space is the real line and $k = 1$, the optimal solution is the median and the mean respectively. In the MATROID MEDIAN problem, we are given a matroid on the set $\mathbb{F}$, and the set of open facilities must be an independent set in the matroid. Again, the goal is to minimize the assignment cost of clients to the nearest open facility.

In the FACILITY LOCATION problem, an instance is not given k, but additionally has open : $\mathbb{F} \to \mathbb{R}^+$ that indicates the opening cost of each facility. The goal is to find a subset $F \subseteq \mathbb{F}$ (without any restriction on $|F|$) that minimizes $\operatorname{open}(F) + \operatorname{cost}(C, F)$ where $\operatorname{open}(F) := \sum_{f \in F} \operatorname{open}(f)$.

Finally, the *aspect ratio* of a metric space (V, d) is $\Delta := \frac{\max_{x, y \in V} d(x, y)}{\min_{x, y \in V} d(x, y)}$.

2   The Approximation Algorithm

We now give the $(1 + 2/e + \varepsilon)$-approximation algorithm for k-Median, where $\varepsilon > 0$ is a fixed parameter throughout this section. The running time of the algorithm is $f(k, \varepsilon) \cdot \mathrm{poly}(n)$, where $f(k, \varepsilon) = O(\varepsilon^{-2} k \log k)^k$. We then indicate the alterations to get algorithms for k-Means and Matroid Median.

2.1   The Intuition

We focus on k-Median for now; the ideas for the other problems are analogous. The first idea is to reduce the size of the client set C to $O(\varepsilon^{-2} k \log n)$– this can be done by results on core-sets for k-Median, which consolidate the clients into a small number of distinct locations [8, 14]. The consolidated clients now have weights, but this extension to weighted k-Median does not pose a problem.

The next idea is to carefully enumerate over the structure of an optimal solution. Consider an optimal solution $F^\star = \{f_1^\star, \ldots, f_k^\star\}$. For a facility $f_i^\star \in F^\star$, let "cluster" $C_i^\star$ be the clients assigned to $f_i^\star$, i.e., the subset of clients C for which $f_i^\star$ is closest open facility. Let ℓ_i be the client in $C_i^\star$ closest to $f_i^\star$ – we call it the *leader* of cluster $C_i^\star$. Let R_i be the distance $d(f^\star, \ell_i)$, suitably discretized. Our algorithm guesses the leaders ℓ_i and the distances R_i for each $i \in [k]$. Since the size of C is $O(\varepsilon^{-2} k \log n)$, there are $(O(\varepsilon^{-2} k \log n))^k$ choices for leaders,[1] and a similar number of choices for the distances; moreover, this quantity can be shown to be $f(k, \varepsilon) \cdot n^{O(1)}$.

Assume now that we have correctly guessed the leaders ℓ_i and distances R_i. For each leader ℓ_i, let F_i be the facilities at distance about R_i from ℓ_i – this set F_i contains $f_i^\star$. By making copies, assume the sets F_i are disjoint. Now our task is to select one facility from each set F_i such that the total (weighted) assignment cost of the clients in C is minimized. As such, this seems like a decreasing *supermodular minimization* problem with a (partition) matroid constraint. (Observe that choosing an arbitrary center in each F_i gives us a 3-approximation in FPT time, but we want to do much better.)

The last idea is to convert this into a monotone submodular maximization problem, again with a partition matroid constraint. For each set F_i, we add a *fictitious facility* f_i' such that (i) the assignment cost of clients to the fictitious facilities is at most $3OPT$, and (ii) for a subset S of facilities, the "improvement" $\mathsf{cost}(C, F') - \mathsf{cost}(C, F' \cup S)$, where F' is the set of fictitious facilities, is a monotone submodular function. We finally show that a $(1 - 1/e)$-approximation for this submodular maximization problem gives the desired approximation guarantee. The next two sections describe the algorithm for k-Median in detail. The extension to k-Means, Matroid Median and Facility Location then appears in the full version of the paper.

2.2   Client Reduction via Coresets

Consider an instance $\mathcal{I} = ((V, d), C, \mathbb{F}, k)$ of the k-Median problem. Let $\varepsilon > 0$ be a fixed constant. We now define the notion of core-sets and use known results to reduce the size of C to (a weighted) a set of size $O(\varepsilon^{-2} k \log n)$.

---

[1] Our analysis will tighten this bound to $O(\varepsilon^{-2} \log n)^k$, but this improvement can be ignored for this intuition section.

▶ **Definition 5** (Core-set). *A* (strong) core-set *for $\mathcal{I}$ is a set of clients $C' \subseteq V$ along with weights w_j for all $j \in C'$, such that*

$$\sum_{j \in C'} w_j \, d(j, F) \in (1 - \varepsilon, 1 + \varepsilon) \cdot \sum_{j \in C} d(j, F),$$

for every $F \subseteq \mathbb{F}$ with $|F| = k$.

A similar definition holds for a strong core-set for the k-MEANS problem. Since we deal only with strong core-sets in this paper, we drop the modifier and refer to them only as core-sets. The first core-sets for metric k-MEDIAN were given by Chen [8]; the following result is the best current construction:

▶ **Theorem 6** ([14], Theorem 15.4). *For $0 \le \varepsilon, \delta \le 1/2$, there exists a Monte Carlo algorithm that for each instance I of k-MEDIAN on a general metric, outputs a core-set $C' \subseteq C$ with size*

$$|C'| = O\Big(\frac{k \log n + \log 1/\delta}{\varepsilon^2} \Big)$$

with probability $1 - \delta$, where $n = |V|$. Moreover, the algorithm runs in time $O(k(n + k) + \log^2(1/\delta) \log^2 n)$. For k-MEANS, the core-set is of size $|C'| = O\big(\frac{k \log n + \log 1/\delta}{\varepsilon^4} \big)$, and the runtime remains the same.

The power of core-sets lies in the following fact.

▷ Fact 7.   Consider a k-MEDIAN/k-MEANS instance $\mathcal{I} = ((V, d), C, \mathbb{F}, k)$, and let C' be a (strong) core-set with weights w. Consider the weighted instance $\mathcal{I}' = ((V, d), C', \mathbb{F}, k, w)$, which is the instance $\mathcal{I}$ with its clients replaced by the weighted clients in the core-set. Then, for any $\beta \ge 1$, a β-approximate solution $F \subseteq \mathbb{F}$ to $\mathcal{I}'$ is a $\beta(1 + O(\varepsilon))$-approximate solution to $\mathcal{I}$.

Therefore, in order to find a $(1 + 2/e + O(\varepsilon))$-approximation to a k-MEDIAN $\mathcal{I}$, it suffices to find a $(1 + 2/e + O(\varepsilon))$-approximation to $\mathcal{I}'$, and analogously for k-MEANS. Henceforth, we restrict our attention to the core-set instance $\mathcal{I}'$. In other words, we assume that our instances have only a small number of clients, but now the clients have associated weights. In the following sections, we show how to approximate such weighted k-MEDIAN/k-MEANS instances in FPT time.

2.3   Reduction to Submodular Maximization

Given Fact 7, we only consider instances $\mathcal{I} = ((V, d), C, \mathbb{F}, k, w)$ of weighted k-MEDIAN, where clients in C have weights in the range $[1, n]$ and $|C|$ is bounded by $O(\varepsilon^{-2} k \log n)$. In this section we prove the following approximation guarantee for k-MEDIAN; this, combined with Fact 7, proves the k-MEDIAN statement in Theorem 1.

▶ **Theorem 8.** *Let ε be a fixed parameter. Given a k-MEDIAN instance $\mathcal{I} = ((V, d), C', \mathbb{F}, k, w)$ with $|C'| = O(\varepsilon^{-2} k \log n)$, there is a $(1 + 2/e + O(\varepsilon))$-approximation algorithm that runs in $f(k, \varepsilon) n^{O(1)}$ time.*

By scaling, assume the minimum distance between points in V is 1, so the aspect ratio Δ is the maximum distance between two points in V. For a positive integer a, define $\llbracket a \rrbracket := (1 + \varepsilon)^{\lceil \log_{(1+\varepsilon)} a \rceil}$ as the smallest power of $(1 + \varepsilon)$ larger than or equal to a. Here, ε is the same fixed parameter as the one used in the core-set.

The formal algorithm follows the intuition in §2.1 and is described in Algorithm 2.1; let us step through it now. We iterate over all possible values $\ell_1, \ldots, \ell_k$ for the leaders, and $R_1, \ldots, R_k$ for the corresponding distances. The same vertex could appear several times in the subset $\{\ell_1, \ldots, \ell_k\}$, and so the latter should be thought of as a multi-set. In Step 7, we add k new fictitious facilities: for each i, the new facility f'_i is at distance $2R_i$ from all the facilities in F_i. The distance to all other points is determined by triangle inequality in Step 8. Claim 9 shows that this forms a valid metric. In Step 9, we define the "improvement" function $\mathsf{improv}(S)$ as the reduction in cost due to adding in the facilities in S. Claim 10 shows this function is monotone submodular. This means we can use the $(1 - 1/e)$-approximation algorithm [4] for monotone submodular function maximization subject to a matroid constraint to find a set S which contains exactly one facility from each of the sets F_i, since this is a partition matroid constraint. Observe that the function $\mathsf{improv}(\cdot)$ can be computed efficiently. This completes the description of the algorithm.

Algorithm 2.1 FindCenters.

1: **for** every multi-set $\{\ell_1, \ell_2, \ldots, \ell_k\} \subseteq C$ **do**
2:     **for** every $R_1, \ldots, R_k$ such that $R_i \in [1, \ldots, [\![\Delta]\!]]$ and R_i is a power of $(1 + \varepsilon)$ **do**
3:         $F_i \leftarrow \{f \in F \mid [\![d(f, \ell_i)]\!] = R_i\}$
4:         make copies of facilities to ensure that $F_1, \ldots, F_k$ are disjoint
5:         Initialize $F' \leftarrow \varnothing$; F' will be the set of fictitious facilities
6:         **for** $i = 1, \ldots, k$ **do**
7:             add a new fictitious facility f'_i to F' and set $d(f'_i, f) := 2R_i$ for all $f \in F_i$
8:             define $d(f'_i, v) := \min_{f \in F_i}(d(f'_i, f) + d(f, v))$ for all other points v
9:         define $\mathsf{improv}(S) := \mathsf{cost}(C, F') - \mathsf{cost}(C, S \cup F')$ for every set $S \subseteq \mathbb{F}$
10:         find $S \subseteq \mathbb{F}$ approximately maximizing $\mathsf{improv}(S)$, such that $|S \cap F_i| = 1$
11: among all sets S computed in line 10, output S for which $\mathsf{cost}(C, S)$ is minimized.

To prove correctness of the algorithm, we need to show two things: the distance function defined on $F' \cup V$ in Step 8 is a metric, and the function improv defined in Step 9 is monotone and submodular. We defer the simple proofs to the full version of the paper.

▷ **Claim 9.** Consider the set F' defined during an iteration of the algorithm. The distance function defined on $F' \cup V$ is a metric.

▷ **Claim 10.** The function $\mathsf{improv}(S)$ defined in Step 9 is monotone and submodular with $\mathsf{improv}(\varnothing) = 0$.

Now to bound the runtime. Since $|C| = O(\varepsilon^{-2}k \log n)$, there are at most $\binom{O(\varepsilon^{-2}k \log n) + k - 1}{k} = (O(\varepsilon^{-2} \log n))^k$ different multi-sets of size k with elements in C. In addition, there are $\log_{1+\varepsilon} \Delta$ many choices for R_i for each $i \in [k]$. Therefore, the number of iterations in Step 1 of the algorithm can be bounded by

$$(O(\varepsilon^{-2} \log n))^k \cdot (\log_{1+\varepsilon} \Delta)^k \leq \left(O\Big(\frac{(\log \Delta)(\log n)}{\varepsilon^2}\Big) \right)^k. \tag{2.1}$$

Since we started with the unweighted k-MEDIAN problem, the aspect ratio Δ can be assumed to polynomially bounded in n, and so the number of iterations can be bounded by $(O(\log n/\varepsilon^2))^k$, which is at most $n \cdot (O(\varepsilon^{-2}k \log k))^k$. Indeed, in case $k < \frac{\log n}{\log \log n}$, $(O(\log n/\varepsilon^2))^k \leq (O(1/\varepsilon^2))^k \cdot (\log n)^{\frac{\log n}{\log \log n}} = (O(1/\varepsilon^2))^k \cdot n$. Else $\log n \leq O(k \log k)$, and hence $(O(\log n/\varepsilon^2))^k = (O(k \log k/\varepsilon^2))^k$.

The algorithm for submodular maximization subject to a matroid constraint takes polynomial time, given a value oracle for the function [4, Theorem 1.1]: in fact it can be sped up for the case of partition matroid constraints [4, §3.3]. The value oracle for $\mathsf{improv}(S)$ can itself be implemented in polynomial time. Hence each iteration of the algorithm can be run in time polynomial in n.

The submodular maximization algorithm is a randomized Monte-Carlo algorithm that succeeds with only probability $1 - 1/n^2$, but we can easily boost the success probability by repetition: by running it $\tau := \mathrm{poly}(\varepsilon^{-1} k \log n)$ times for each input S and returning the maximum value obtained, we can ensure that with high probability it succeeds in all the calls we make.

2.3.1   Approximation Ratio

We now argue about the approximation ratio of the algorithm. We fix an optimal solution to the instance. Let $F^\star = \{f_1^\star, \ldots, f_k^\star\}$ be the centers opened by this solution. Define $C_i^\star$ as the clients for which the closest open center is $f_i^\star$, i.e., $C_i^\star := \{j \in C : d(j, f_i^\star) = d(j, F^\star)\}$. We define the notion of leaders with respect to this solution.

▶ **Definition 11** (Leader). *For each $i \in [k]$, call a client $j \in C_i^\star$ that minimizes $d(j, f_i^\star)$ over all $j \in C_i^\star$ the* leader $\ell_i^\star$ *of center $f_i^\star$. If there are multiple clients $j \in C_i^\star$ achieving the minimum, declare an arbitrary one to be the leader. Note that a client can be the leader of multiple centers $f_i^\star$. The* leaders *w.r.t. the solution $F^\star$ is the multi-set $\{\ell_1^\star, \ldots, \ell_k^\star\}$. For each leader $\ell_i^\star$, the* radius $R_i^\star$ *is defined as $\llbracket d(\ell_i^\star, f_i^\star) \rrbracket$.*

Consider the iteration of Algorithm 2.1 where $\ell_1, \ldots, \ell_k$ are equal to $\ell_1^\star, \ldots, \ell_k^\star$ respectively, and $R_1, \ldots, R_k$ are equal to $R_1^\star, \ldots, R_k^\star$ respectively. Let $S^\star$ be the set output in Step 10 of the algorithm. It suffices to show that $\mathsf{cost}(C, S^\star) \le (1 + 2/e + \varepsilon)\mathsf{cost}(C, F^\star)$. We proceed to show this in the rest of the section.

As in the algorithm, define

$$F_i := \{f \in F \mid \llbracket d(f, \ell_i^\star) \rrbracket = R_i^\star\},$$

so that $f_i^\star \in F_i$ for each $i \in [k]$. (Recall that the sets F_i are disjoint by duplicating facilities.) Let $F' = \{f_1', \ldots, f_k'\}$ be the set of fictitious facilities defined in the algorithm.

We are interested in the solutions S that consist of one center from each F_i, since one such solution is the desired $F^\star$. More formally, define a solution S to be *valid* if the set S can be listed as $(f_1, \ldots, f_k)$ so that $f_i \in F_i$ for each $i \in [k]$.

▷ **Claim 12.**   For every valid S, $\mathsf{cost}(C, F' \cup S) = \mathsf{cost}(C, S)$.

Proof. List the set S as $(f_1, \ldots, f_k)$, where $f_i \in F_i$ for each $i \in [k]$. Informally, this claim amounts to showing that the fictitious facilities F' do not improve the solution S. To formalize this idea, fix a client $j \in C$ and a fictitious facility f_i', and let $f \in F_i$ be a closest center to j in F_i. Below, we show that in fact, client j is closer to $f_i \in S$ than to $f_i' \in F'$:

$$d(j, f_i) \overset{(\triangle\ \mathrm{ineq.})}{\le} d(j, f) + d(f, \ell_i^\star) + d(\ell_i^\star, f_i) \le d(j, f) + R_i^\star + R_i^\star = d(j, f) + d(f, f_i') = d(j, f_i').$$

Therefore, we have $d(j, F') \ge d(j, S)$ for all clients j, so

$$\mathsf{cost}(C, F' \cup S) = \sum_{j \in C'} w_j\, d(j, F' \cup S) = \sum_{j \in C'} w_j\, d(j, S) = \mathsf{cost}(C, S),$$

as desired.   ◁

We now bound the cost of the solution which opens facilities at F'.

$\triangleright$ **Claim 13.** $\mathsf{cost}(C, F') \leq (3 + 2\varepsilon)\,\mathsf{cost}(C, F^\star)$.

Proof. It suffices to show that $d(j, F') \leq (3 + 2\varepsilon)\,d(j, F^\star)$ for each client $j \in C$. Fix a client $j \in C$, and let $f_i^\star \in F^\star$ be a center achieving $d(j, f_i^\star) = d(j, F^\star)$. Since $\ell_i^\star$ is the leader of center $f_i^\star$, we have

$$d(j, f_i^\star) \geq d(\ell_i, f_i^\star) \geq \frac{R_i^\star}{1 + \varepsilon}. \tag{2.2}$$

Recall that $f_i^\star \in F_i$. Therefore,

$$d(j, F') \leq d(j, f_i') \overset{(\triangle)}{\leq} d(j, f_i^\star) + d(f_i^\star, f_i') = d(j, f_i^\star) + 2R_i^\star$$
$$\overset{(2.2)}{\leq} d(j, f_i^\star) + 2(1 + \varepsilon)\,d(j, f_i^\star) \leq (3 + 2\varepsilon)\,d(j, F^\star),$$

as desired. $\triangleleft$

Let $S^\star$ be the set output in Step 11. Since the algorithm of [4] is $(1 - 1/e)$-approximation,

$$\mathsf{improv}(S^\star) \geq (1 - 1/e)\mathsf{improv}(F^\star) \tag{2.3}$$

▶ **Lemma 14.** *The solution $S^\star$ in (2.3) satisfies $\mathsf{cost}(C, S^\star) \leq (1 + 2/e + O(\varepsilon))\,\mathsf{cost}(C, F^\star)$.*

Proof. We bound the cost associated with this solution as follows.

$$\mathsf{cost}(C, S^\star) \overset{(\text{Lem } 12)}{=} \mathsf{cost}(F' \cup S^\star) = \mathsf{cost}(C, F') - \mathsf{improv}(S^\star)$$
$$\overset{(2.3)}{\leq} \mathsf{cost}(C, F') - (1 - 1/e)\,\mathsf{improv}(F^\star)$$
$$= \mathsf{cost}(C, F') - (1 - 1/e)\,(\mathsf{cost}(C, F') - \mathsf{cost}(C, F^\star))$$
$$= (1/e)\,\mathsf{cost}(C, F') + (1 - 1/e)\,\mathsf{cost}(C, F^\star)$$
$$\overset{(\text{Lem } 13)}{\leq} (3 + 2\varepsilon)(1/e)\,\mathsf{cost}(C, F^\star) + (1 - 1/e)\,\mathsf{cost}(C, F^\star) \tag{2.4}$$
$$= (1 + 2/e + O(\varepsilon))\,\mathsf{cost}(C, F^\star).$$

Hence the proof. ◀

2.3.2 Putting it all together

Our algorithm is a Monte Carlo randomized algorithm: both our subroutines use randomness. The first is the core-set construction in §2.2, and the second is the submodular maximization procedure in Step 10 of the algorithm. For each, we can make the error probability $1/\mathsf{poly}(n)$. Since each iteration of the algorithm can be implemented in $\mathsf{poly}(n)$ time, the runtime is dominated by the number of iterations, which is $(O(\varepsilon^{-2}k \log k)^k \mathsf{poly}(n))$. Moreover, combining the two steps of finding the core-set and the submodular maximization, the approximation ratio is $(1 + \varepsilon)(1 + 2/e + O(\varepsilon)) = 1 + 2/e + O(\varepsilon)$. This proves Theorem 1 for the k-MEDIAN problem.

3 Gap-ETH Hardness of Max k-Coverage

In this section, we show that assuming the Gap Exponential Time Hypothesis (Gap-ETH) [12, 24], for any $\varepsilon > 0$, there is no FPT-approximation algorithm that approximates MAX k-COVERAGE better than a factor $(1 - 1/e + \varepsilon)$.

▶ **Theorem 15** (Hardness for Max-Coverage). *There exists a function $g : \mathbb{R}^+ \to \mathbb{R}^+$ such that assuming the Gap-ETH, for any $\varepsilon > 0$, any $(1 - 1/e + \varepsilon)$-approximation algorithm for MAX k-COVERAGE with n elements and m sets must run in time at least $(n + m)^{k^{g(\varepsilon)}}$.*

Using the reduction of Guha and Khuller [16], this immediately implies Theorem 2. The rest of the section is devoted to the proof of Theorem 15. The proof has two main components: the first part shows under the Gap-ETH, it takes at least $n^{h(k)}$ time to approximate the LABEL COVER problem even when one side of the bipartition has only k vertices; here $h(\cdot)$ is some increasing function depending on the quality of approximation. This reduction is inspired by the recent progress on the hardness of parameterized problems [5, 10] and was communicated to us by Pasin Manurangsi. The second part is the classical reduction from LABEL COVER to MAX k-COVERAGE given by Feige [13].

3.1 Hardness of Label Cover from Gap-ETH

We begin with the standard definition of LABEL COVER.

▶ **Definition 16** (Label Cover). *An instance of LABEL COVER $\mathcal{L}$ consists of a bipartite graph $G = (U \cup V, E)$ with possibly parallel edges, two label sets Σ_U, Σ_V, and a projection $\pi_e : \Sigma_U \to \Sigma_V$ for each $e \in E$. Given a labeling $\sigma : (U \cup V) \to (\Sigma_U \cup \Sigma_V)$, an edge $e = (u, v) \in E$ is satisfied when $\pi_e(\sigma(u)) = \sigma(v)$. The goal of LABEL COVER is to find a labeling σ that maximizes the number of satisfied edges. Let $\mathsf{OPT}(\mathcal{L})$ be the maximum fraction of edges simultaneously satisfied by any labeling.*

Note that we include the *projection property* in the definition; all LABEL COVER instances in the paper will have this property. For a vertex $u \in U \cup V$, let d_u be the degree of u, and let d_U (resp. d_V) be the maximum degree of U (resp. V). We also call an instance *U-regular* (resp. *V-regular*) if all vertices in U (resp. V) have the same degree. All subsequent LABEL COVER instances will be U-regular, though the lack of V-regularity will require us to do a little more work in §3.2.

Given a 3-SAT formula ϕ, let $\mathsf{OPT}(\phi)$ be the maximum fraction of clauses that can be satisfied by any assignment. The Gap-ETH [12, 24] states that there exist some constants $\delta > 0, s < 1$ for which no algorithm, given a 3-SAT formula ϕ on n variables and $m = O(n)$ clauses, can distinguish whether $\mathsf{OPT}(\phi) = 1$ or $\mathsf{OPT}(\phi) < s$ in time $O(2^{\delta n})$. The main result of this subsection is the following lemma.

▶ **Lemma 17.** *For every $\ell, r \in \mathbb{N}$, there is a reduction that, given 3-SAT formula ϕ with n variables and m clauses, outputs a U-regular LABEL COVER instance $\mathcal{L}$ such that*
- *(Completeness) $\mathsf{OPT}(\phi) = 1 \implies \mathsf{OPT}(\mathcal{L}) = 1$, and*
- *(Soundness) $\mathsf{OPT}(\phi) < s \implies \mathsf{OPT}(\mathcal{L}) < s^{\Omega(r)}$,*

where $|U| = \ell^r, |V| = n^r, |\Sigma_U| = 2^{O(mr/\ell)}, |\Sigma_V| = 2^{O(r)}, d_V \leq m^r$. The running time of this reduction is $m^{O(r)} \cdot |\Sigma_U|$.

In particular, assuming Gap-ETH, for any $\eta > 0$, if we let $r = \Theta(\log(1/\eta))$ so that

$$|\Sigma_U|^{|U|^{O(1/\log(1/\eta))}} = |\Sigma_U|^{|U|^{1/2r}} = |\Sigma_U|^{\ell^{1/2}} = 2^{O(mr/\sqrt{\ell})},$$

no algorithm can take a LABEL COVER instance $\mathcal{L}$ and can decide whether $\mathsf{OPT}(\mathcal{L}) = 1$ or $\mathsf{OPT}(\mathcal{L}) < \eta$ in time $|\Sigma_U|^{|U|^{O(1/\log(1/\eta))}}$.

Note that a brute-force algorithm that tries every assignment to U and chooses the best assignment for V for it runs in $O(|\Sigma_U|^{|U|})$ times a polynomial. Lemma 17 shows that assuming the Gap-ETH, even approximately solving LABEL COVER requires significant time.

Lemma 17 is proved by a series of well-known transformations between LABEL COVER instances. We start with the following basic hardness result for LABEL COVER assuming the Gap-ETH, which follows from essentially restating Gap-ETH as a *clause-variable* game:

▶ **Theorem 18** (Theorem 4.1 of [5])**.** *There is a reduction that, given 3-SAT formula ϕ with n variables and m clauses, outputs a U-regular* LABEL COVER *instance $\mathcal{L}$ such that*
- *(Completeness)* $\mathsf{OPT}(\phi) = 1 \implies \mathsf{OPT}(\mathcal{L}) = 1$, *and*
- *(Soundness)* $\mathsf{OPT}(\phi) < s' \implies \mathsf{OPT}(\mathcal{L}) < s = 1 - (1 - s')/3$,

where $|U| = m, |V| = n, |\Sigma_U| = 7, |\Sigma_V| = 2, d_V \le m$, and G is U-regular with $d_U = 3$. In particular, assuming the Gap-ETH, there exist constants $\delta > 0$, $s < 1$ such that no algorithm can take a LABEL COVER *instance $\mathcal{L}$ and can decide whether $\mathsf{OPT}(\mathcal{L}) = 1$ or $\mathsf{OPT}(\mathcal{L}) < s$ in $O(2^{\delta|U|})$ time.*

Let ℓ be a parameter that will be related to k in MAX k-COVERAGE later. We can ensure ℓ divides $|U|$ by taking an arbitrary vertex in U and making $\ell\lceil|U|/\ell\rceil - |U|$ copies of it. This does not change any of the properties in Theorem 18 except to increase the soundness s by $o_n(1)$; however, the soundness still remains bounded away from 1.

Since we want few vertices on the left, we construct a new LABEL COVER instance $\mathcal{L}_1$ by partitioning U into ℓ groups and creating super-vertices for each one. Formally, index the vertices of u as $U = \{u_{i,j}\}_{i \in [\ell], j \in [m/\ell]}$, and let the ith part be $S_i := \{u_{i,j}\}_{j \in [m/\ell]}$. The new instance $\mathcal{L}_1 = ((U_1 \cup V_1, E_1), \Sigma_{U_1}, \Sigma_{V_1}, \{\pi_e^1\}_{e \in E_1})$ is constructed as follows.

- $V_1 = V$ and $\Sigma_{V_1} = \Sigma_V$ (the RHS remains unchanged),
- $U_1 = \{S_1, \ldots, S_\ell\}$. $\Sigma_{U_1} = (\Sigma_U)^{m/\ell}$ (the LHS has one super-vertex for each group), and
- for each $e = (u, v) \in E$ such that $u = u_{i,j}$, add an edge $e' = (S_i, v)$ to E_1 with the projection $\pi_{e'}^1(\sigma_1, \ldots, \sigma_{m/\ell}) := \pi_e(\sigma_j)$ where the latter π_e denotes the projection in $\mathcal{L}$. (Recall we allow parallel edges with different projections.)

Since the set of possible labelings and the set of edges remain the same except for syntactic changes, the completeness c and the soundness s do not change. The parameters become $|U_1| = \ell, |V_1| = |V|, |\Sigma_{U_1}| = 2^{O(m/\ell)}, |\Sigma_{V_1}| = O(1)$. It still maintains U-regularity and $d_{V_1} = d_V \le m$.

The final transformation is the powerful *parallel repetition* step, which shows that the soundness decreases exponentially as we take the natural graph power. Fix $r \in \mathbb{N}$. The instance $\mathcal{L}_2 = ((U_2 \cup V_2, E_2), \Sigma_{U_2}, \Sigma_{V_2}, \{\pi_e^2\}_{e \in E_2})$ is constructed as follows.
- $U_2 = (U_1)^r$ and $\Sigma_{U_2} = (\Sigma_{U_1})^r$.
- $V_2 = (V_1)^r$ and $\Sigma_{V_2} = (\Sigma_{V_1})^r$.
- $E_2 = (E_1)^r$. For each $e = (e_i)_{i \in [r]} \in E_2$ with $e_i = (u_i, v_i) \in E_1$ and $(\sigma_1, \ldots, \sigma_r) \in \Sigma_{U_1}^r$, $\pi_e^2(\sigma_1, \ldots, \sigma_r) = (\pi_{e_1}^1(\sigma_1), \ldots, \pi_{e_r}^1(\sigma_r))$.

The parameters become $|U_2| = \ell^r, |V_2| = |V|^r = n^r, |\Sigma_{U_2}| = 2^{O(mr/\ell)}, |\Sigma_{V_2}| = 2^{O(r)}, d_{V_2} \le d_{V_1}^r \le m^r$, and $\mathcal{L}_2$ maintains U-regularity. The completeness c still remains 1, and by the parallel repetition theorem [25], the soundness drops $s = 2^{-\Theta(r)}$, where the constant hiding in the Θ depends on the original soundness. This proves Lemma 17.

3.2 Hardness of Max k-Coverage from Label Cover

Given the "nice" LABEL COVER instance from Lemma 17 we now show how to reduce this to MAX k-COVERAGE. This reduction is standard and closely follows the classical one given by Feige [13], modulo some minor issues arising from it not being V-regular.

Recall that an instance of $\mathcal{I}$ of MAX k-COVERAGE consists of an underlying universe $\mathcal{U}$, a family $\mathcal{S}$ of subsets, and an integer k. The goal is to find a subfamily $\mathcal{S}' \subseteq \mathcal{S}$ with $|\mathcal{S}'| = k$ that covers the largest number of elements. For notational simplicity, we prove the hardness

of the *weighted version* of Max k-Coverage where each element $e \in \mathcal{U}$ has weight $w(e)$ and we want to maximize the total weight of the covered elements. Note that weighted instances can be easily converted to unweighted instances by duplicating elements according to their weights. In our reduction, the ratio between the maximum and the minimum weight will be bounded by the number of elements. The proof appears in the full version of the paper.

▶ **Lemma 19** (Reduction #2). *There exist functions $a : \mathbb{R}^+ \to \mathbb{N}$ and $f : \mathbb{R}^+ \to \mathbb{R}^+$ such that for any $\varepsilon > 0$, there exists a polynomial-time reduction that takes a* Label Cover *instance $\mathcal{L} = ((U \cup V, E), \Sigma_U, \Sigma_V, \{\pi_e\}_{e \in E})$ that is U-regular and has the maximum V-degree d_V, and produces a* Max k-Coverage *instance $\mathcal{I} = (\mathcal{U}, \mathcal{S}, k)$ such that*

- *(Completeness)* $\mathsf{OPT}(\mathcal{L}) = 1 \implies \mathsf{OPT}(\mathcal{I}) = w(\mathcal{U})$.
- *(Soundness)* $\mathsf{OPT}(\mathcal{L}) < f(\varepsilon) \implies \mathsf{OPT}(\mathcal{I}) \leq (1 - 1/e + \varepsilon) \cdot w(\mathcal{U})$.

The reduction satisfies $|\mathcal{U}| \leq |V| \cdot |d_V|^{a(\varepsilon)} \cdot a(\varepsilon)^{\Sigma_V}$, $|\mathcal{S}| = a(\varepsilon) \cdot |U| \cdot |\Sigma_U|$, and $k = a|U|$.

We can now finish the proof of Theorem 15 based on Lemma 17 and Lemma 19.

Proof of Theorem 15. Fix $\varepsilon > 0$ that determines $a(\varepsilon)$ and $f(\varepsilon)$ in Lemma 19. Let $r \in \mathbb{N}$ in Lemma 17 so that the soundness $2^{-\Omega(r)} \leq f(\varepsilon)$.

With ℓ still being a free parameter, Lemma 17 shows a reduction from an initial 3-SAT instance ϕ with n variables and $m = O(n)$ clauses to a Label Cover instance with $|U| = \ell^r, |V| = n^r, |\Sigma_U| = 2^{O(mr/\ell)}, |\Sigma_V| = 2^{O(r)}$, and $d_V \leq m^r$. Lemma 19 with this Label Cover instance produces a Max k-Coverage instance with

$$|\mathcal{U}| \leq |V| \cdot |d_V|^a \cdot a^{\Sigma_V} = n^{O(ar)} \cdot a^{2^{O(r)}}$$
$$|\mathcal{S}| = a \cdot |U| \cdot |\Sigma_U| = a\ell^r \cdot 2^{O(nr/\ell)}$$
$$k = a|U| = a\ell^r.$$

An $(1 - 1/e + \varepsilon)$-approximation algorithm for Max k-Coverage that runs in time $|\mathcal{S}|^{k^{1/2r}}$ will distinguish whether $\mathsf{OPT}(\phi) = 1$ or $\mathsf{OPT}(\phi) < s'$ for some s' in time

$$2^{O((nr/\ell) \cdot k^{1/2r})} = 2^{O((nr/\ell) \cdot \sqrt{\ell})} = 2^{O(nr/\sqrt{\ell})},$$

which will contradict the Gap-ETH for large enough ℓ. Observe that $|\mathcal{S}| \gg |\mathcal{U}|$; if we set $g(\varepsilon) := 1/2r$, we get the same implication from an algorithm that runs in time $|\mathcal{U}|^{k^{g(\varepsilon)}}$, which proves the theorem.   ◀

References

1   Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better Guarantees for k-Means and Euclidean k-Median by Primal-Dual Algorithms. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 61–72, 2017.

2   Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The Hardness of Approximation of Euclidean k-Means. In *31st International Symposium on Computational Geometry, SoCG 2015, June 22-25, 2015, Eindhoven, The Netherlands*, pages 754–767, 2015.

3   Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 737–756. SIAM, 2014.

4   Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011. `doi:10.1137/080733991`.

5   Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-ETH to FPT-inapproximability: Clique, dominating set, and more. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 743–754. IEEE, 2017.

6   Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A Constant-Factor Approximation Algorithm for the k-Median Problem. *J. Comput. Syst. Sci.*, 65(1):129–149, 2002. `doi:10.1006/jcss.2002.1882`.

7   Ke Chen. On k-Median clustering in high dimensions. In *SODA*, 2006.

8   Ke Chen. On coresets for k-median and k-means clustering in metric and Euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.

9   Yijia Chen and Bingkai Lin. The constant inapproximability of the parameterized dominating set problem. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 505–514. IEEE, 2016.

10   Karthik C.S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1283–1296. ACM, 2018.

11   Artur Czumaj and Christian Sohler. Small Space Representations for Metric Min-sum k-Clustering and Their Applications. *Theory Comput. Syst.*, 46(3):416–442, 2010. `doi:10.1007/s00224-009-9235-1`.

12   Irit Dinur. Mildly exponential reduction from gap 3SAT to polynomial-gap label cover. *Electronic Colloquium on Computational Complexity (ECCC)*, pages TR 16–128, 2016.

13   Uriel Feige. A Threshold of Ln N for Approximating Set Cover. *J. ACM*, 45(4), July 1998.

14   Dan Feldman and Michael Langberg. A Unified Framework for Approximating and Clustering Data. *CoRR*, abs/1106.1379, 2011. (An extended abstract appeared at STOC 11). `arXiv:1106.1379`.

15   Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, PCA and projective clustering. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1434–1453. Society for Industrial and Applied Mathematics, 2013.

16   Sudipto Guha and Samir Khuller. Greedy strikes back: improved facility location algorithms. *J. Algorithms*, 31(1):228–248, 1999. `doi:10.1006/jagm.1998.0993`.

17   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. `doi:10.1016/j.comgeo.2004.03.003`.

18   Ravishankar Krishnaswamy, Amit Kumar, Viswanath Nagarajan, Yogish Sabharwal, and Barna Saha. Facility Location with Matroid or Knapsack Constraints. *Math. Oper. Res.*, 40(2):446–459, 2015. `doi:10.1287/moor.2014.0678`.

19   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. `doi:10.1145/1667053.1667054`.

20   Euiwoong Lee, Melanie Schmidt, and John Wright. Improved and simplified inapproximability for k-means. *Inf. Process. Lett.*, 120:40–43, 2017. `doi:10.1016/j.ipl.2016.11.009`.

21   Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Information and Computation*, 222:45–58, 2013.

22   Shi Li and Ola Svensson. Approximating k-Median via Pseudo-Approximation. *SIAM J. Comput.*, 45(2):530–547, 2016. `doi:10.1137/130938645`.

23   Bingkai Lin. The parameterized complexity of k-biclique. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 605–615. SIAM, 2015.

24 Pasin Manurangsi and Prasad Raghavendra. A Birthday Repetition Theorem and Complexity of Approximating Dense CSPs. In *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 78:1–78:15, 2017.

25 Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

26 Chaitanya Swamy. Improved Approximation Algorithms for Matroid and Knapsack Median Problems and Applications. *ACM Trans. Algorithms*, 12(4):49:1–49:22, 2016. `doi:10.1145/2963170`.