Approximation algorithms for $k$-median with lower-bound constraints

Ameet Gadekar
Department of Computer Science, Aalto University, Finland

Bruno Ordozgoiti
Department of Computer Science, Aalto University, Finland

Suhas Thejaswi
Department of Computer Science, Aalto University, Finland

Abstract
We study a variant of the classical $k$-median problem known as diversity-aware $k$-median (introduced by Thejaswi et al. 2021), where we are given a collection of facility subsets, and a solution must contain at least a specified number of facilities from each subset. We investigate the fixed-parameter tractability of this problem and show several negative hardness and inapproximability results, even when we afford exponential running time with respect to some parameters of the problem.

Motivated by these results we present a fixed parameter approximation algorithm with approximation ratio $(1 + \frac{2}{3} + \epsilon)$, and argue that this ratio is essentially tight assuming the gap-exponential time hypothesis. We also present a simple, practical local-search algorithm that gives a bicriteria $(2k, 3 + \epsilon)$ approximation with better running time bounds.

2012 ACM Subject Classification Mathematics of computing → Combinatorial algorithms; Theory of computation → Fixed parameter tractability; Theory of computation → Facility location and clustering

Keywords and phrases Algorithmic fairness, $k$-median, clustering, parameterized algorithms, parameterized approximation algorithms

Digital Object Identifier 10.4230/LIPIcs...
Approximation algorithms for \( k \)-median with lower-bound constraints

1 Introduction

In the \( k \)-median problem (\( k \)-MEDIAN) we are given two sets of points, clients and facilities, embedded in a metric space. The goal is to pick \( k \) facilities to minimize the sum of distances from the clients to their closest facility. In a now classic paper, Arya et al. \[4\] showed that a simple local-search heuristic achieves a \((3 + \epsilon)\)-approximation for this problem. Currently, the best known approximation ratio stands at 2.611 \[2\]. On the other side of the coin, the \( k \)-MEDIAN problem is known to be \( \text{NP} \)-hard to approximate to a factor less than \( 1 + 2/e \) \[8\]. Bridging this gap is a well known open problem.

In recent years the attention has turned in part to variants of the problem with constraints on the solution. One such variant is the red-blue median problem (\( rb \)-MEDIAN), in which the facilities are colored either red or blue, and a solution may contain only up to a specified number of facilities of each color \[10\]. This formulation was generalized by the matroid-median problem (\( \text{MATROIDMEDIAN} \)) \[12\], where solutions must be independent sets of a matroid. Constant-factor approximation algorithms were given by Hajiaghayi et al. \[10, 9\] and Krishnaswamy et al. \[12\] for \( rb \)-MEDIAN and \( \text{MATROIDMEDIAN} \) problems, respectively.

In this paper we consider yet another variant, the diversity-aware \( k \)-median problem (\( \text{DIV}-k\)-MEDIAN), which was proposed recently by Thejaswi et al. in the context of algorithmic fairness \[10\]. In this formulation, in addition to the sets of clients \( C \) and facilities \( F \), we are also given a collection \( \mathcal{G} = \{G_1, \ldots, G_t\} \subseteq 2^F \) of facility groups, and corresponding lower-bound requirements \( \{r_1, \ldots, r_t\} \subset \mathbb{N} \). A solution to the \( \text{DIV}-k\)-MEDIAN problem is required to contain at least \( r_i \) facilities from group \( G_i \), while simultaneously minimizing the \( k \)-MEDIAN clustering objective.

In contrast to the clustering variants discussed earlier, the \( \text{DIV}-k\)-MEDIAN problem admits no polynomial-time approximation algorithms. The reason, as shown by Thejaswi et al. \[16\], is simply that the problem of finding a feasible solution is itself \( \text{NP} \)-hard. Given the hopeless prospects of tackling this problem with polynomial-time algorithms, in this paper we investigate the potential of \textit{fixed-parameter tractable} (\( \text{FPT} \)) algorithms. That is, can we solve the problem, either exactly or approximately, if we limit the combinatorial explosion in running time to a set of input parameters?

To motivate the choice of parameters for designing \( \text{FPT} \) algorithms, we characterize the hardness of \( \text{DIV}-k\)-MEDIAN based on standard complexity theory assumptions. Observe that the \( \text{DIV}-k\)-MEDIAN problem is an amalgamation of two independent problems: (i) finding a subset of facilities \( S \subseteq F \) of size \(|S| = k\) that satisfies the requirements \(|S \cap G_i| \geq r_i\) for all \( i \in [t]\), and (ii) minimizing the \( k \)-MEDIAN clustering cost. To remain consistent with the problem statement of Thejaswi et al. \[16\], we refer to subproblem (i) as the \( \text{DIV}-k\)-MEDIANq problem, where the cost of clustering is ignored. If we ignore the requirements in (i) we obtain the classical \( k \)-MEDIAN formulation, which immediately establishes the \( \text{NP} \)-hardness of \( \text{DIV}-k\)-MEDIAN.

A reduction of the vertex cover problem to \( \text{DIV}-k\)-MEDIANq is sufficient to show that \( \text{DIV}-k\)-MEDIAN is inapproximable to any multiplicative factor in polynomial-time even if all the subsets are of size two \[15\, \text{Theorem 3}\]. The \( \text{W}[2]\)-hardness of \( \text{DIV}-k\)-MEDIAN with respect to parameter \( k \) is a consequence of the fact that \( k \)-MEDIAN is \( \text{W}[2]\)-hard, which follows from a reduction by Guha and Kuller \[8\]. More strongly, combining the result of \[16\, \text{Lemma 1}\] with the strong exponential time hypothesis (\( \text{SETH} \)) \[11\], we arrive at the following conclusions: if we only consider the parameter \( k \), a trivial exhaustive-search algorithm is our best hope for finding an optimal, or even an approximate, solution to \( \text{DIV}-k\)-MEDIAN.

\[\textbf{Corollary 1.} \text{Assuming } \text{SETH. For all } k \geq 3 \text{ and } \epsilon > 0, \text{ there exists no } O((|F|)^{k-\epsilon})\]
algorithm to find an optimal solution for Div-$k$-Median, furthermore, there exists no \( \Theta((|F|)^{k-\epsilon}) \) algorithm to approximate Div-$k$-Median to any multiplicative factor.

Given the \( \mathbf{W}[2] \)-hardness of Div-$k$-Median with respect to parameter $k$, it is natural to consider other relaxations of the problem. An obvious question is whether we can approximate Div-$k$-Median in FPT time with respect to $k$, if we are allowed to open, say, $f(k)$ facilities instead of $k$, for some function $f$. Unfortunately, this is also unlikely as Div-$k$-Median captures $k$-DomSet [16] Lemma 1, and even finding a dominating set of size $f(k)$ is \( \mathbf{W}[1] \)-hard [13].

\textbf{Proposition 2.} For any function $f(k)$, finding $f(k)$ facilities that approximate the Div-$k$-Median cost to any multiplicative factor in FPT time with respect to parameter $k$ is \( \mathbf{W}[1] \)-hard.

A possible way forward is to identify other parameters of the problem, to design FPT algorithms to solve the problem optimally. As established earlier, $k$-Median is a special case of Div-$k$-Median when $t = 1$. This immediately rules out an exact FPT algorithm for Div-$k$-Median with respect to parameters $(k, t)$. Furthermore, we caution the reader against entertaining the prospects of other, arguably natural, parameters, such as the maximum lower bound $r_{\max} = \max_{i \in [t]} r[i]$ (ruled out by the relation $r[i] \leq k$) and the maximum number of groups a facility can belong to $\mu = \max_{f \in F}(|G_i : f \in G_i, i \in [t]|)$ (ruled out by the relaxation $\mu \leq t$).

\textbf{Proposition 3.} Finding an optimal solution for Div-$k$-Median is \( \mathbf{W}[2] \)-hard with respect to parameters $(k, t)$, $(r_{\max}, t)$ and $(\mu, t)$.

The above intractability results thwart our hopes of solving the Div-$k$-Median problem optimally in FPT time. We then ask, are there any parameters of the problem that allow us to find an approximate solution in FPT time? We answer this question positively, and present a tight FPT-approximation algorithm with respect to parameters $(k, t)$, that is, the number of facilities chosen $k$ and the number of facility groups $t$.

1.1 Our results

Our main result, stated below, shows that a constant-factor approximation of Div-$k$-Median can be achieved in FPT time with respect to parameters $(k, t)$. In fact, somewhat surprisingly, the factor is the same as the one achievable for $k$-Median. So despite the stark contrast in their polynomial-time approximability, the FPT landscape is rather similar for these two problems. We also note that under the gap-exponential time hypothesis (Gap-ETH), the approximation ratio achieved in Theorem 4 is essentially tight for any FPT algorithm with respect to $(k, t)$. This follows from combining the fact that the case $t = 1$ is essentially $k$-Median with a result of Cohen-Addad et al. [5], which assuming Gap-ETH gives a lower bound for any FPT algorithm with respect to $k$. This bound matches their —and our— approximation guarantee.

\textbf{Theorem 4.} For every $\epsilon > 0$, there exists a randomized $(1 + \frac{2}{\epsilon} + \epsilon)$-approximation algorithm for the Div-$k$-Median problem which runs in time $f(k, t, \epsilon) \cdot \text{poly}(|U|)$, where

\[
f(k, t, \epsilon) = \Theta \left( \left( \frac{2^t k^3 \log^2 k}{\epsilon^2 \log(1+\epsilon)} \right)^k \right).
\]

Furthermore, the approximation ratio is tight for any FPT algorithm with respect to $(k, t)$, assuming Gap-ETH.

Finally, in Section 1.4 we will point out a simple observation: by relaxing the upper bound on the number of facilities to at most $2k$, we can use a practical local-search heuristic and obtain a slightly weaker quality guarantee with better running time bounds.
Remark 5. For every $\epsilon > 0$, there exists a randomized $(3 + \epsilon)$-approximation algorithm that outputs at most $2k$ facilities for the Div-$k$-MEDIAN problem in time $f(k, t) \cdot \text{poly}(|U|, 1/\epsilon)$, where $f(k, t) = O(2^{kt})$.

1.2 Our techniques

To achieve the result stated in Theorem 4, first we partition the facility set $F$ into $2^t$ disjoint subsets. The facilities in each subset share the same characteristic vector, which encodes the groups they belong to. More precisely, the characteristic vector of a facility $f$ is a vector of length $t$ such that its $i$-th index is set to 1 if $f \in G_i$, 0 otherwise. Next, we enumerate over all $k$-multisets of the partition, but retain only those corresponding to a feasible solution - meaning, picking one facility from each subset of such $k$-multiset satisfies the lower-bound requirements. To this end, we leverage a combinatorial structure of the partition to find such $k$-multisets in time $O(2^{kt}\text{poly}(|U|))$. Finally, we show that at least one such $k$-multiset is guaranteed to contain an optimal set of facilities, provided a feasible solution exists. In order to achieve this, for each $k$-multiset containing a feasible solution we create an instance of $k$-MEDIAN-$k$-PM$^1$, where the $k$-multiset determines the partition matroid constraint. At this point, the stage will be set to use the FPT algorithm of Cohen-Addad et al. [5], which approximately solves $k$-MEDIAN, to achieve the approximation factor of Theorem 4.

On the other hand, Theorem 5 follows from the observation that, if we pad our facility set to satisfy the lower-bound requirements, we can employ the classic result of Arya et al. [1] to bound the local-search approximation factor. Note that as observed by Thejaswi et al. [16, Section 5.1], the local-search algorithm has an unbounded locality gap in Div-$k$-MEDIAN, so this trick only applies to FPT algorithms. Bounding the approximation factor of local search restricted to $k$-facilities is left as an open problem.

The rest of the paper is structured as follows: in Section 2 we review related work, and continue to introduce notation and formal definition of the problems in Section 3. In Section 4, we present our main algorithmic results.

2 Related work

In this section we present a brief survey of related work.

The Div-$k$-MEDIAN problem was recently introduced by Thejaswi et al. [16]. They gave inapproximability results for the general case where facility groups may intersect, and presented constant-factor approximation algorithms for tractable cases where facility groups are disjoint. For the case where disjoint facility groups and the number of facility groups is two (red-blue), they presented a $5 + \epsilon$ approximation by reducing the problem to $rb$-MEDIAN and using a local-search heuristic by Hajiaghayi et al. [9]. For the more general variant with any number of disjoint facility groups, they presented an 8-approximation algorithm based on linear programing. The algorithm relies on a reduction to MatroidMedian and builds on the algorithm by Swamy et al. [14]. For the most general case, with possibly intersecting facility groups, they presented heuristics based on local search. To the best of our knowledge, there is no known algorithm to solve the general variant of Div-$k$-MEDIAN with theoretical guarantees.

---

1 Roughly, $k$-MEDIAN-$k$-PM is $k$-MEDIAN with $k$-partition matroid constraint. Refer Section 3.2 for formal definition.
The $k$-Median clustering formulation is a classic problem in computer science. The first constant-factor approximation for metric $k$-Median was presented by Charikar et al. [4], which was improved to $(3 + \epsilon)$ in a now seminal work by Arya et al. [1], using a local-search heuristic. The best known approximation ratio for metric instances stands at 2.611, which is due to Byrka et al. [2]. In the FPT landscape, finding an optimal solution for $k$-Median is known to be $W[2]$-hard with respect to parameter $k$ due to a reduction by Guha and Khuller [8]. More recently, Cohen-Addad et al. [5] presented FPT approximation algorithms with respect to parameter $k$, with a $(1 + \frac{3}{k} + \epsilon)$ approximation ratio. They showed that the ratio is essentially tight assuming Gap-ETH. Their result also implies a $(2 + \epsilon)$ approximation algorithm for MATROID-Median in FPT time with respect to parameter $k$.

### 3 Preliminaries

In this section we introduce our notation and formal problem definitions.

#### 3.1 Notation

Given a metric space $(U, d)$, a set $C \subseteq U$ of clients, a set $F \subseteq U$ of facilities and a subset $S \subseteq F$ of facilities, we denote by $\text{cost}(S) = \sum_{c \in C} d(c, S)$ the clustering cost of $S$, where $d(c, S) = \min_{s \in S} d(c, s)$. We say that $C$ is weighted when every $c \in C$ is associated to a weight $w_c \in \mathbb{R}$, and the clustering cost becomes $\text{cost}(S) = \sum_{c \in C} w_c \cdot d(c, S)$. Similarly, for $C' \subseteq C$ and $S \subseteq F$, we write $\text{cost}(C', S) = \sum_{c \in C'} w_c \cdot d(c, S)$. Further, given a collection $\mathcal{F} = \{G_1, \ldots, G_t\}$ of facility groups such that $G_i \subseteq F$, for each facility $f \in F$ we denote by $\vec{\chi}_f \in \{0, 1\}^t$ the characteristic vector of $f$ with respect to $\mathcal{F}$. Said vector is obtained by setting the index $\vec{\chi}_f[i] = 1$ if $f \in G_i$, 0 otherwise, for all $i \in [t]$. For $\eta > 0$ and $a \in \mathbb{Z}_{\geq 0}$, we denote $[a]_{\eta} \in \mathbb{Z}$ as the smallest integer such that $(1 + \eta)^{[a]_{\eta}} \geq a$.

In this paper we make use of standard parameterized complexity terminology. We refer the interested reader to the book by Cygan et al. [6].

#### 3.2 Problem definitions

We start by formally defining the diversity-aware $k$-median problem.

**Definition 6 (Diversity-aware $k$-median ($\text{DIV}-k$-Median)).** We are given a metric space $(U, d)$, a set $C \subseteq U$ of clients, a set $F \subseteq U$ of facilities, a collection $\mathcal{F} = \{G_1, \ldots, G_t\}$ of facility sets $G_i \subseteq F$ (called groups), a budget $k \leq |F|$ and a vector of requirements $\vec{r} = (r[1], \ldots, r[t])$, that is, one threshold $r[i] \leq k$ for each group $G_i$. The problem asks to find a subset of facilities $S \subseteq F$ of size $k$, satisfying $|S \cap G_i| \geq r_i$ for all $i \in [t]$, such that the clustering cost of $S$, $\text{cost}(S) = \sum_{c \in C} d(c, S)$ is minimized. An instance of DIV-$k$-Median is specified as $I = ((U, d), F, C, \mathcal{F}, \vec{r}, k)$.

Another problem of our interest is a variant of $k$-Median with additional $p$-partition matroid constraints, which we now formally define.

**Definition 7 ($k$-Median with $p$-Partition Matroid ($\text{DIV}-k$-Median-$p$-PM)).** We are given a metric space $(U, d)$, a set of clients $C \subseteq U$, a set of facilities $F \subseteq U$ and a collection $\mathcal{E} = \{E_1, \ldots, E_p\}$ of disjoint facility groups called a $p$-partition matroid. The problem asks to find a subset of facilities $S \subseteq F$ of size $k$, containing at most one facility from each group $E_i$, so that the clustering cost of $S$, $\text{cost}(S) = \sum_{c \in C} d(c, S)$ is minimized. An instance of $k$-Median-$p$-PM is specified as $I = ((U, d), F, C, \mathcal{E}, k)$. 


4 Algorithms

In this section we present FPT approximation algorithms for the Div-$k$-Median problem. For this section, by FPT, we mean FPT w.r.t. $(k,t)$.

Our algorithms work as follows: first, we carefully enumerate, in FPT time, collections of facility subsets which potentially satisfy the lower-bound requirements (Section 4.1). For each existing feasible solution we obtain a constant-factor approximation (Section 4.3). Since at least one of these feasible solutions is optimal — provided the instance is feasible in the first place — the corresponding approximate solution will be an approximate solution to the Div-$k$-Median problem. The pseudocode is available in Algorithm I.

Additionally, we show (Section 4.4) that by padding the facility set to satisfy the lower-bound requirements, we can employ a local-search heuristic — or any other approximation algorithm — to obtain a bicriteria approximation algorithm. This approach, in particular, returns at most $2k$ facilities with a $(3 + \epsilon)$-approximation factor.

4.1 Finding feasible constraint patterns

To begin, let us clarify what a constraint pattern is. Recall that in the Div-$k$-Median problem we are given a collection of possibly intersecting facility groups $\mathcal{G} = \{G_1, \ldots, G_k\}$. We associate each facility $f \in F$ with a characteristic vector $\vec{\chi}_f \in \{0,1\}^k$, such that the index $\vec{\chi}_f[i]$ is set to 1 if $f \in G_i$, 0 otherwise, for all $i \in [k]$. For each $\vec{\gamma} \in \{0,1\}^k$ let $E(\vec{\gamma}) = \{f : \vec{\chi}_f = \vec{\gamma}, f \in F\}$ denote the set of all facilities with characteristic vector $\vec{\gamma}$.

Finally, $\mathcal{E} = \{E(\vec{\gamma}) : \vec{\gamma} \in \{0,1\}^k\}$ is a set of mutually disjoint facility groups. Indeed, $\mathcal{E}$ induces a partition on $F$.

Given a $k$-multiset $\mathcal{E} = \{E(\vec{\gamma}_1), \ldots, E(\vec{\gamma}_k)\} \subset \mathcal{E}$, the constraint pattern associated to $\mathcal{E}$ is the vector obtained by the element-wise sum of the characteristic vectors $\{\vec{\gamma}_1, \ldots, \vec{\gamma}_k\}$. That is, $\sum_{j \in [k]} \vec{\gamma}_j$. A constraint pattern is said to be feasible if $\sum_{j \in [k]} \vec{\gamma}_j \geq \vec{r}$, where the inequality is taken element-wise.

Lemma 8. Given an instance $I = ((U,d), C, F, \mathcal{G}, k)$ of Div-$k$-Median, we can enumerate all the $k$-multisets with feasible constraint pattern in time $O(2^k \text{poly}(|U|))$ time.

Proof. There are $\binom{|\mathcal{E}| + k - 1}{k}$ possible $k$-multisets of $\mathcal{E}$, so enumerating feasible constraint patterns can be done in $O(|\mathcal{E}|^k |U|)$ time. The size of the partition $|\mathcal{E}|$ is bounded by $\min\{n, 2^k\}$, since there can be at most $\min\{n, 2^k\}$ unique characteristic vectors. So the time complexity of enumerating all feasible constraint patterns is $O(2^k \text{poly}(|U|))$. ▶

Observe that for every $k$ multiset $\mathcal{E} = \{E(\vec{\gamma}_1), \ldots, E(\vec{\gamma}_k)\}$ with a feasible constraint pattern, picking an arbitrary facility from each $E(\vec{\gamma}_j)$ results in a feasible solution to the Div-$k$-Median instance $I$.

Before moving to FPT-approximation of the Div-$k$-Median problem, it is convenient to introduce the concept of coresets.

4.2 Coresets

Our algorithm relies on the notion of coresets. At a high-level, the idea is to reduce the number of clients in a $k$-Median-$k$-PM instance such that the distortion in the sum of distances is bounded to a multiplicative factor $(1 + \nu)$, for some $\nu > 0$. Given an instance $((U,d), C, F, k)$ of the k-Median-$k$-PM problem, for every $\nu > 0$ we can reduce the number of clients in $C$ to a weighted set $C' = \mathcal{O}(\nu^{-2}k \log |U|)$. Towards this end we make use of the coreset construction for $k$-Median by Feldman and Langberg [7] and extend the
approach to solve the $k$-Median-$k$-PM problem. To our knowledge, this is the best known framework for constructing coresets.

- **Definition 9 (Coreset).** Given an instance $I = ((U, d), C, F, k)$ of the $k$-Median problem and a constant $\nu > 0$, a (strong) coreset is a subset of clients $C' \subseteq C$ with associated weights $\{w_c : c \in C'\}$ such that for any subset of facilities $S \subseteq F$ of size $|S| = k$ it holds that

$$\sum_{c \in C'} \sum_{c \in C} d(c, S) \leq (1 + \nu) \cdot \sum_{c \in C} w_c \cdot d(c, S)$$

- **Theorem 10 ([7], Theorem 4.9).** Given a metric instance $I = ((U, d), C, F, k)$ of the $k$-Median problem, for each $\nu > 0$, $\delta < \frac{1}{2}$, there exists a randomized algorithm that, with probability at least $1 - \delta$, computes a coreset $C' \subseteq C$ of size $|C'| = \Theta(\nu^{-2}(k \log |U| + \log \frac{1}{\delta}))$ in time $\Theta(k(\log |F| + k) + \log^2 \frac{1}{\delta} \log^2 |U|)$.

## 4.3 FPT approximation algorithms

In this section we present our main result. We will first give an intuitive overview of our algorithm. As a warm-up, we will describe a simple $(3 + \epsilon)$-approximation algorithm. Then, we will show how to obtain a better guarantee, leveraging recent FPT techniques for $k$-Median.

### Intuition

Given an instance $I = ((U, d), C, F, k)$ of Div-$k$-MEDIAN, we first partition the facility set $F$ into $2^k$ subsets $\mathcal{E} = \{E(\vec{\gamma}) : \vec{\gamma} \in \{0, 1\}^k\}$, such that each subset $E(\vec{\gamma})$ corresponds to all the facilities whose characteristic vector is $\vec{\gamma} \in \{0, 1\}^k$. Then, using Lemma 5 we enumerate all $k$-multisets with feasible constraint pattern. For each feasible $k$-multiset $\mathcal{E} \subset \mathcal{E}$, we generate an instance $I_E$ of $k$-Median-$k$-PM, resulting in at most $O(2^{2k})$ instances. Next we build a coreset $C' \subseteq C$ of clients, for which we rely on Theorem 10. In our final step, we obtain an approximate solution to $k$-Median-$k$-PM adapting the techniques from [5], which we discuss below.

Let $\mathcal{E} = \{E(\vec{\gamma}_1), \ldots, E(\vec{\gamma}_k)\} \subset \mathcal{E}$ be a $k$-multiset with a feasible constraint pattern, and let $J_\mathcal{E}$ be the corresponding feasible $k$-Median-$k$-PM instance. Let $F^* = \{f^*_i \in E(\vec{\gamma}_i)\}_{i \in [k]}$ be an optimal solution of the instance $J_\mathcal{E}$. For each $f^*_i$, let $c^*_i \in C'$ be a closest client, with $d(f^*_i, c^*_i) = \lambda^*_i$. Next, for each $c^*_i$ and $\lambda^*_i$, let $\Pi^*_i \subseteq E(\vec{\gamma}_i)$ be the set of facilities $f \in E(\vec{\gamma}_i)$ such that $d(f, c^*_i) = \lambda^*_i$. Observe that, for each $i \in [k]$, $\Pi^*_i$ contains $f^*_i$ of the optimal solution $F^*$. Thus, if only we knew $c^*_i$ and $\lambda^*_i$ for all $i \in [k]$, we would be able to obtain a provably good solution.

To find closest client $c^*_i$ and its corresponding distance $\lambda^*_i$, we employ techniques due to Cohen-Addad et al. [5], which they build on the work of Feldman and Langberg [7]. The idea is to reduce the search spaces small enough to allow brute-force search in FPT time. This is accomplished as follows: Firstly, to reduce number of clients we use client coreset $C'$, so we have $|C'| = O(k\nu^{-2}\log |U|)$. Hence, enumerating all ordered $k$-multisets of $C'$ can be done in $O((k\nu^{-2}\log |U|)^k)$ time. For bounding the search space of $\lambda^*_i$ (which is at most $\Delta^k = |U|^\Omega(k)$), we can discretize the range $[\Delta]$ to $[\Delta]^\eta$, for some $\eta > 0$. Note that $[\Delta]_\eta \leq \log_{1+\eta} \Delta = O(\log |U|)$. Hence, enumerating all ordered $k$-multisets of $[\Delta]_\eta$, we spend at most $O(\log^2 |U|)$ time.

Using the facilities in $\{\Pi^*_i\}_{i \in [k]}$, we find an approximate solution for the instance $J_\mathcal{E}$. First, as a warm-up, we show in Lemma 11 that picking exactly one facility, arbitrarily from each $\Pi^*_i$ already gives a $(3 + \epsilon)$ approximate solution. Finally, in Lemma 12 we obtain a better
Approximation algorithms for $k$-median with lower-bound constraints

Figure 1 An illustration of facility selection for FPT algorithm for solving $k$-MEDIAN-$k$-PM instance, based on the ideas of Cohen-Addad et al. [5].

- **Lemma 11.** For every $\epsilon > 0$, there exists a randomized $(3 + \epsilon)$-approximation algorithm for the DIV-$k$-MEDIAN problem which runs in time $f(k,t,\epsilon) \cdot \text{poly}(|U|)$, where $f(k,t,\epsilon) = O((2^t \epsilon^{-2} k^2 \log k)^k)$.

**Proof.** Consider an instance of $k$-MEDIAN-$k$-PM $J = ((U,d),\{C \cup E(\vec{c}_1^*,\ldots,\vec{c}_k^*)\},k)$, corresponding to an optimal solution. That is, for some optimal solution $S^* = \{f_1^*,\ldots,f_k^*\}$, we have $f_j^* \in E(\vec{c}_j^*)$. We consider, in addition, the iteration where the chosen clients and radii are optimal, i.e. $d(c_j^*,f_j^*) = \lambda_j^*$ is minimal over all clients served by $f_j^*$ in the optimal solution. The construction is illustrated in Figure 1.

We define $B_i$ to be the set of facilities in $E(\vec{c}_i^*)$ at a distance of at most $\lambda_i^*$ from $c_i^*$. We will now argue that picking one arbitrary facility from each $B_i$ gives a 3-approximation with respect to an optimal pick. In more detail, a solution for $k$-MEDIAN-$k$-PM must pick exactly one facility from each $E(\vec{c}_j^*)$ for all $j \in [k]$. Let $F^* = \{f_1^*,\ldots,f_k^*\}$ be an optimal solution such that $f_j^* \in E(\vec{c}_j^*)$ for each $j \in [k]$. Let $C_j^* \subseteq C'$ be a set of clients assigned to each facility $f_j^*$ in optimal solution. Let $\{f_1,\ldots,f_k\}$ be the arbitrarily chosen facilities, such that $f_j \in B_j$. Then for any $c \in C_j$

$$d(c,f_j) \leq d(c,f_j^*) + d(f_j^*,c_j) + d(c_j,f_j).$$

By the choice of $c_j^*$ we have $d(f_j^*,c_j^*) + d(c_j,f_j) \leq 2\lambda_j^*$, which implies

$$\sum_{c \in C_j} d(c,f_j) \leq 3 \sum_{c \in C_j} d(c,f_j^*).$$

By the properties of the coreset and bounded discretization error [5], we obtain the approximation stated in the lemma. ▶

We will now focus on our main result, stated in Theorem 4. As stated before, we build upon the ideas for $k$-MEDIAN of Cohen-Addad et al. of [6]. Their algorithm, however, does not apply directly to our setting, as we have to ensure that the chosen facilities satisfy the constraints.
A key observation is that by relying on the partition-matroid constraint of the auxiliary submodular optimization problem, we can ensure that the output solution will satisfy the constraint pattern. Since at least one constraint pattern contains an optimal solution, we obtain the advertised approximation factor.

In the following lemma we argue that this is indeed the case. Next, we will provide an analysis of the running time of the algorithm. This will complete the proof of Theorem 4.

Lemma 12. On input \((U, d), (E^*_1, \ldots, E^*_k), C^o\), Algorithm 2 outputs a set \(S\) satisfying cost\((S) \leq (1 + \frac{2}{k} + \epsilon)\)cost\((S^*)\).

Proof. Let \(S^* = \{f^*_1, \ldots, f^*_k\}\) be an optimal solution to Div-\(k\)-Median. By definition, \(f^*_i \in E^*_i\). Consider the iteration where the chosen clients and radii are optimal, that is, \(\lambda^*_i = d(e^*_i, f^*_i)\) and this distance is minimal over all clients served by \(f^*_i\) in the optimal solution.

Assuming the input described in the statement of the lemma, it is clear that in this iteration we have \(f^*_i \in \Pi_i\) (see Algorithm 2, line 5). Furthermore, given the partition-matroid constraint imposed on it, the proposed submodular optimization scheme is guaranteed to pick exactly one facility from each of \(\Pi_i\), for all \(i\).

On the other hand, known results for submodular optimization show that this problem can be efficiently approximated to a factor within \((1 - 1/e)\) of the optimum \([3]\). It is not difficult to see this translates into a \((1 + \frac{2}{k} + \epsilon)\)-approximation of the optimal choice of facilities, one from each of \(\Pi_i\), \([3]\). This concludes the proof.

Running Time: First we bound the running time of Algorithm 2. Note that, the runtime of Algorithm 2 is dominated by the two for loops (Line 2 and 3), since remaining steps, including finding approximate solution to the submodular function, runs in time \(\text{poly}(|U|)\). The for loop of clients (Line 2) takes time \(\mathcal{O}(k\nu^{-2}\log |U|^k)\). Similarly, the for loop of discretized distances (Line 3) takes time \(\mathcal{O}((|\Delta|_i)^k) = \mathcal{O}((\log k_i + \gamma |U|)^k)\), since \(\Delta = \text{poly}(|U|)\). Hence, setting \(\eta = \Theta(\epsilon)\), the overall running time of Algorithm 2 is bounded by

\[
\mathcal{O}\left(\left(\frac{k\log^2 |U|}{\epsilon^2 \log(1 + \epsilon)}\right)^k \text{poly}(|U|)\right) = \mathcal{O}\left(\left(\frac{k^3 \log^2 k}{\epsilon^2 \log(1 + \epsilon)}\right)^k \text{poly}(|U|)\right).
\]

Since, Algorithm 1 invokes Algorithm 2 for \(\mathcal{O}(2^k)\) times, its running time is bounded by

\[
\mathcal{O}\left(\left(\frac{2^k k^3 \log^2 k}{\epsilon^2 \log(1 + \epsilon)}\right)^k \text{poly}(|U|)\right).
\]

4.4 A bicriteria FPT approximation algorithm

Finally, we describe a straightforward approach to obtain a bicriteria approximation algorithm. This relies on the following simple observation. We can first run our scheme for obtaining feasible constraint patterns in order to find a feasible solution. Then, we can just ignore the lower-bound constraints and run any polynomial-time approximation algorithm for \(k\)-Median. By taking the union of the two solutions, we obtain at most \(2k\) facilities which satisfy the

\[\text{cost}(S) \leq (1 + \frac{2}{k} + \epsilon)\text{cost}(S^*)\]
Approximation algorithms for \( k \)-median with lower-bound constraints

**Algorithm 1** \( \text{Div-}k\text{-Med}(I = ((U, d), F, C, \{G_1, \cdots, G_t\}, k, \epsilon)) \)

**Input:** \( I \), an instance of the \( \text{Div-}k\text{-Median} \) problem

**Output:** \( T^* \), subset of facilities

1. \( \text{foreach } \vec{\gamma} \in \{0, 1\}^t \) do

2. \( E(\vec{\gamma}) \leftarrow \{ f \in F : \vec{\gamma} = \vec{\chi}_f \} ; \quad \text{// facilities with characteristic vector } \vec{\gamma} \)

3. \( \mathcal{E} \leftarrow \{ E(\vec{\gamma}) : \vec{\gamma} \in \{0, 1\}^t \} ; \quad \text{// partition of facilities} \)

4. \( C' \leftarrow \text{coreset}((U, d), F, C, k, \nu \leftarrow \epsilon/27) \)

5. \( T^* \leftarrow \emptyset \)

6. \( \text{foreach multiset } \{ E(\vec{\gamma}_1), \cdots, E(\vec{\gamma}_k) \} \subseteq \mathcal{E} \text{ of size } k \) do

7. \( \text{if } \sum_{i \in [k]} \vec{\gamma}_i \geq \vec{r}, \text{ element-wise then} \)

8. \( T \leftarrow k\text{-Median-PM}((U, d), \{ E(\vec{\gamma}_1), \cdots, E(\vec{\gamma}_k) \}, C', \epsilon/2) \)

9. \( \text{if } \text{cost}(C', T) < \text{cost}(C', T^*) \) then

10. \( T^* \leftarrow T \)

11. \( \text{return } T^* \)

**Algorithm 2** \( k\text{-MedPM}(J = ((U, d), \{ E_1, \cdots, E_k \}, C', \epsilon') \)

**Input:** \( J \), an instance of the \( k\text{-Median-k-PM} \) problem

**Output:** \( S^* \), a subset of facilities

1. \( S^* \leftarrow \emptyset \)

2. \( \text{foreach ordered multiset } \{ c'_1, \cdots, c'_k \} \subseteq C' \text{ of size } k \) do

3. \( \text{foreach ordered multiset } \Lambda = \{ \lambda_1, \cdots, \lambda_k \} \) such that \( \lambda_i \subseteq [\Delta_{ij}] \) do

4. \( \text{for } j = 1 \text{ to } k \) do

5. \( \Pi_j \leftarrow \{ f \in E'_j \mid d_D(f, c'_j) = \lambda_i \} \)

6. \( \text{Add a fictitious facility } F'_j \)

7. \( \text{for } f \in \Pi_j \) do

8. \( d(F'_j, f) \leftarrow 2\lambda_i \)

9. \( \text{for } f \notin \Pi_j \) do

10. \( d(F'_j, v) \leftarrow 2\lambda_i + \min_{f \in \Pi_j} d(f, v) \)

11. \( \text{for } S \subseteq F, \text{ define } \text{improve}(S) := \text{cost}(C', F') - \text{cost}(C', F' \cup S) \) do

12. \( S_{max} \leftarrow S \subseteq F \text{ that maximizes } \text{improve}(S) \) s.t. \( |S \cap \Pi_j| = 1, \forall j \in [k] \)

13. \( \text{if } \text{cost}(C', S_{max}) < \text{cost}(C', S^*) \) then

14. \( S^* \leftarrow S \)

15. \( \text{return } S^* \)
lower-bound constraints and achieve the quality guarantee of the chosen approximation algorithm (with respect to the optimal solution of size $k$).

We can employ, for instance, the local-search heuristic of Arya et al., which yields a $(2k, 3 + \epsilon)$-approximation \cite{ary2004}, or the result of Byrka et al. \cite{byr2014} to achieve a $(2k, 2.611)$-approximation.

---

**References**

\[1\] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for $k$-median and facility location problems. *SIAM Journal on computing*, 33(3):544–562, 2004.

\[2\] Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for $k$-median, and positive correlation in budgeted optimization. In *SODA*, pages 737–756. SIAM, 2014.

\[3\] Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

\[4\] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the $k$-median problem. *JCSS*, 65(1):129–149, 2002.

\[5\] Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT Approximations for $k$-Median and $k$-Means. In *ICALP*, volume 132, pages 42:1–42:14, Dagstuhl, Germany, 2019. Dagstuhl.

\[6\] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 4. Springer, 2015.

\[7\] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *STOC*, page 569–578. ACM, 2011.

\[8\] Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. In *SODA*, page 649–657, USA, 1998. SIAM.

\[9\] M Hajiaghayi, Rohit Khandekar, and Guy Kortsarz. Local-search algorithms for the red-blue median problem. *Algorithmica*, 63(4):795–814, 2012.

\[10\] MohammadTaghi Hajiaghayi, Rohit Khandekar, and Guy Kortsarz. Budgeted red-blue median and its generalizations. In *ESA*, 2010.

\[11\] Russell Impagliazzo and Ramamohan Paturi. On the complexity of $k$-sat. *JCSS*, 62(2):367–375, 2001.

\[12\] Ravishankar Krishnaswamy, Amit Kumar, Viswanath Nagarajan, Yogish Sabharwal, and Barna Saha. The matroid median problem. In *SODA*, 2011.

\[13\] Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. *Journal of ACM*, 66(5), 2019.

\[14\] Chaitanya Swamy. Improved approximation algorithms for matroid and knapsack median problems and applications. *ACM Trans. of Algorithms*, 12(4), 2016.

\[15\] Suhas Thejaswi, Bruno Ordozgoiti, and Aristides Gionis. Diversity-aware $k$-median: Clustering with fair center representation, 2021. [arXiv:2106.11696](https://arxiv.org/abs/2106.11696).

\[16\] Suhas Thejaswi, Bruno Ordozgoiti, and Aristides Gionis. Diversity-aware $k$-median: Clustering with fair center representation. In *ECML-PKDD*, pages 1–16. Springer, 2021.