Diversity-aware Clustering: Computational Complexity and Approximation Algorithms

Suhas Thejaswi[1*], Ameet Gadekar[2*], Bruno Ordozgoiti[3] and Aristides Gionis[4]

[1*]Max Planck Institute for Software Systems.
[2]Department of Computer Science, Bar-Ilan University.
[4]Division of Theoretical Computer Science, KTH Royal Institute of Technology.

*Corresponding author(s). E-mail(s): thejaswi@mpi-sws.org; ameet.gadekar@biu.ac.il;
Contributing authors: bruno.ordozgoiti@gmail.com; argioni@kth.se;

Abstract

Keywords: Algorithmic fairness, Fair clustering, Intersectionality, Subgroup fairness

1 Introduction

Diversity is an essential design choice across numerous real-world contexts, spanning social environments [1], organizational structures [2], and demographic studies [3]. Embracing diversity entails acknowledging and incorporating multifaceted characteristics within groups. This concept holds profound relevance in addressing real-world challenges, particularly in scenarios where *intersectionality* — the interconnected nature of social categorizations such as gender, ethnicity, religion, socio-economic status and sexual orientation — plays a pivotal role [4, 5].

Consider the task of constituting a representative committee that accurately mirrors the demography of a broader population. In the pursuit of diversifying, and recognizing its significance in the context of fairness, it is imperative to ensure representation from various groups based on their gender, ethnicity, and economic status, among other [6]. In reality, individuals belong to multiple social categories, for example, a person could be a woman of a specific ethnic background and economic group.

Focusing solely on gender may oversight the representation with respect to ethnicity and economic status. Furthermore, considering the groups independently could neglect the intersectionality of these identities, resulting in a limited understanding of an individual's experiences. In an empirical study, Kearns et al. [7] show that intersectional subgroups may experience increased algorithmic harm. Despite fair algorithmic outcomes be observed individually for gender and race, disparities may emerge at their intersection. Notably, the outcome of algorithmic classifiers differ significantly between black and white females [8].

The task of choosing a diverse committee can be formulated as a clustering problem. In such a scenario, the distance between individuals serves as a measure of (dis)similarity of their view points. At the same time, individuals are associated with attributes such as gender, ethnicity, and economic status, forming groups. The goal is to identify a subset of individuals where the inclusion of minimum number of individuals from each group is ensured, while simultaneously minimizing the distance between the chosen individuals and the broader population. The optimization task aims to strike a balance between group representation and overall cohesion in the committee-formation process.

Currently, algorithms designed to ensure fairness by addressing representation for each group independently, do not consider the complexity introduced by intersecting attributes. Consequently, fairness measures applied separately to each group may not capture the nuanced ways in which biases manifest when multiple attributes are considered together, as highlighted in earlier works [7–11]. For instance, if committee members are selected based on gender, ethnicity, and economic status independently, there is a risk of overlooking the cumulative impact of biases in the intersection of these attributes. On the other hand, an algorithm that accounts for the interconnected nature of these attributes will be better equipped to form diverse and inclusive solutions, acknowledging the complex interplay between multiple protected attributes.

This paper focuses into the exploration of *diversity-aware clustering* problems, aiming to tackle the intricacies in real-world scenarios, where attributes associated with entities or individuals from intersecting groups. By formulating these problems within the clustering framework, our endeavor is to provide comprehensive insights into the complexities arising from intersectionality and to devise methodologies for handling diversity in data clustering, where the cluster centers encapsulate intersectionality.

In diversity-aware clustering, we are given a set of data points in a metric space, a collection of potentially intersecting groups of data points, a specified lower-bound indicating the minimum number of points to select from each group, and the desired number of cluster centers to be chosen. The task is to find a subset of data points of desired size that fulfills the lower-bound criterion for each group while simultaneously minimizing a clustering objective. This paper focuses on three common clustering objectives: *k-median*, *k-means*, and *k-supplier*. The k-median objective seeks to minimize the sum of distances between data points and their closest cluster center. Similarly, the k-means objective aims to minimize the sum of squared distances between data points and their closest cluster center. Finally, the k-supplier objective minimizes the maximum distance between the data points and their closest cluster center. Building upon these clustering objectives, we define diversity-aware k-median, diversity-aware k-means, and diversity-aware k-supplier problems, respectively. For a precise formulation of the problems, see Section 3.

In relevant literature, the data points are distinguished as *clients* and *facilities*, which may or may not be disjoint, and the cluster centers must be chosen exclusively from facilities. Guided by this notion, a distinction is made between *k-center* and *k-supplier*. In the former, all data points are considered potential facilities, and consequently, are eligible to be chosen as cluster centers, whereas in the latter, only a subset of data points are considered as facilities [12].

We distinguish diversity-aware clustering from *fair clustering* formulations based on whether the groups are intersecting or mutually disjoint. Specifically, in case of disjoint groups, we refer to the corresponding problems as *fair-k-median*, *fair-k-means*, and *fair-k-supplier*.

To solve fair-clustering problems, we reduce them to instances of clustering problems with *partition-matroid constraints*. In this setting, the groups are mutually disjoint, and the goal is to select at most one facility from each group, while minimizing the clustering objectives. We define the problems of k-median p-partition matroid, k-means p-partition matroid, and k-supplier p-partition matroid, depending on the clustering objective, with p representing the number of groups. Notably, our observations indicate that solving clustering problems with partition-matroid constraints is comparatively more straightforward and many fair-clustering formulations can be efficiently reduced to these instances. This reduction proves to be pivotal, allowing us to focus on presenting efficient solution for simplified problems.

Clustering is a foundational problem in computer science and is extensively investigated. Various algorithmic outcomes, including computational-complexity results [13, 14], exact exponential algorithms [15], approximation algorithms [16, 17], pseudo-approximation algorithms [18], and parameterized algorithms [14, 19], have been established. The exploration of clustering with fairness aspects has also garnered considerable attention, exploring different notions of fairness [19–24]. Notably, existing research studies on clustering often overlook the possibility of intersecting groups, which is the focus of our work.

1.1 Our contributions

Diversity-aware clustering optimizing the k-median objective with intersecting (facility) groups was introduced by Thejaswi et al. [25], in this setting, constraints are imposed on the minimum number of cluster centers to be chosen from each group. They established the polynomial-time inapproximability for the general variant with intersecting groups and presented polynomial-time approximation algorithms for the case with disjoint groups. In a subsequent work, Thejaswi et al. [26] studied the variant with intersecting groups, offering complexity results and parameterized approximation algorithms for the diversity-aware k-median problem. This paper extends the work of Thejaswi et al. [26], expanding the scope to include prevalent clustering formulations and extending the approach to accommodate both k-means and k-supplier objectives. Additionally, we establish the optimality of presented algorithms based on standard complexity theory assumptions. Furthermore, we improve the approximation ratio of fair-k-median, fair-k-means, and fair-k-supplier, establishing the tightness of approximation ratios. Specifically, our contributions are as follows:

• Aris: Do we want to comment about the lack of experiments? We can say that the extension is theoretical, and experiments can be seen in the previous two conference versions? This could also go to the cover letter.

Parameterized approximation algorithms.

- For diversity-aware k-median, diversity-aware k-means, and diversity-aware k-supplier problems, we present parameterized approximation algorithms with respect to to parameters the number of cluster centers k and groups t, achieving approximation ratios of $1+\frac{2}{e}+\epsilon$, $1+\frac{8}{e}+\epsilon$, and $3+\epsilon$ respectively, where $\epsilon > 0$ (Theorem 6.7 and Theorem 6.13).
- For fair-k-median and fair-k-means, we present parameterized approximation algorithms with respect to parameter k with approximation factors $1+\frac{2}{e}+\epsilon$ and $1+\frac{8}{e}+\epsilon$ (Proposition 7.4).
- For k-median p-partition matroid and k-means p-partition matroid, we present parameterized approximation algorithms with respect to parameter k, achieving approximation factors $1+\frac{2}{e}+\epsilon$ and $1+\frac{8}{e}+\epsilon$ (Corollary 6.9).

Polynomial-time approximation algorithms.

- We improve the polynomial-time approximation ratio of fair-k-supplier from state-of-the-art factor-5 to factor-3 (Proposition 7.5).
- We present a polynomial-time 3-approximation algorithm for the k-supplier k-partition matroid problem (Theorem 6.10).

Computational complexity.

- We study the computational complexity of proposed problems, offering insights into NP-hardness, polynomial-time inapproximability, as well as fixed parameter intractability, that is, when the exponential running time is restricted to specific parameters of the problem (Section 4).
- We establish that the approximation ratios presented for diversity-aware clustering problems are optimal for any parameterized approximation algorithm with respect to parameters k and t. This assertion is grounded based on Gap Exponential Time Hypothesis (Gap-ETH) and parameterized complexity theory assumption that $\mathsf{FPT} \neq \mathsf{W}[2]$ (Theorem 6.7 and Theorem 6.13).
- For fair-k-supplier, we establish that no polynomial-time algorithm can approximate to a factor less than 3, assuming $\mathsf{P} \neq \mathsf{NP}$ and no parameterized algorithm with respect to k can approximate to a factor less than 3, assuming $\mathsf{FPT} \neq \mathsf{W}[2]$ (Observation 7.8).

The subsequent sections of this paper are structured as follows: In Section 2 we discuss the related work. The problem formulations are detailed in Section 3, and Section 4 is dedicated to the discussion of computational complexity results. In Section 6, we present approximation algorithms for diversity-aware clustering, while in Section 7 we present approximation algorithms for fair clustering. Finally, in Section **??** we offer our concluding remarks.

2 Related work

Our work builds on existing literature on clustering problems as well as on recent work on algorithmic fairness.

Clustering. Clustering remains as a fundamental problem in computer science that has been extensively studied, exploring various clustering objectives. This work specifically focuses is on three classical objectives: k-median, k-means, and k-supplier. Further, we emphasize on clustering problems associated to choosing cluster centers, as opposed to the partitioning of data points into clusters. As such, our review of related work is confined to this context.

Charikar et al. [27] presented the first constant factor approximation for k-median in metric spaces, which was improved to $3 + \epsilon$ by Arya et al. [28] using a local-search heuristic. More recently, Cohen-Addad et al. [29] refined the local-search technique to obtain a $2.836 + \epsilon$ approximation. The best-known approximation ratio for metric instances stands at 2.675 due to Byrka et al. [30]. For the k-means problem, Kanungo et al. [31] devised a $9+\epsilon$ approximation algorithm, later improved to 6.357 by Ahmadian et al. [32]. On the other hand, k-median and k-means are NP-hard to approximate to a factor less than $1 + \frac{2}{e}$ and $1 + \frac{8}{e}$, respectively [13]. Bridging the gap between the lower-bound of approximation and the achievable approximation ratio remains a well-known open problem.

In the literature, data points are distinguished as clients and facilities, with a restriction that cluster centers must be exclusively chosen from facilities, allowing the sets of clients and facilities to potentially differ. This distinction gives rise to k-supplier and k-center problems. While both problems aim to minimize the maximum distance between the data points and the closest cluster center, k-supplier requires cluster centers to be chosen exclusively from facilities, whereas k-center allows all data points to be chosen as cluster centers. Variations in complexity results and algorithmic approaches emerge between k-center and k-supplier. For metric k-center, a greedy algorithm that selects farthest data points yields a factor-2 approximation [33], and a factor-3 approximation is known for k-supplier [12]. Assuming $P \neq \mathsf{NP}$, it is NP-hard to approximate k-center to a factor $2 - \epsilon$ [34, Proposition 2] and k-supplier to a factor $3 - \epsilon$, for any $\epsilon > 0$ [12, Theorem 6].

Indeed, designing an (exact) algorithm to find the optimal solution for k-median, k-means, k-center and k-supplier is significantly challenging. *Exhaustive enumeration*, which explores all possible combinations of cluster centers results in algorithm with time $\min\left\{\binom{n}{k}, 2^n\right\} \cdot \mathsf{poly}(n, k)$, where n is the number of data points. For k-center, Fomin et al. [15] presented a non-trivial exact algorithm using subset convolution with time $1.89^n \cdot \mathsf{poly}(n, k)$. Certainly, for k-supplier, an exact algorithm with time $2^{(1-\epsilon)n} \cdot \mathsf{poly}(n, k)$ is unlikely, for any $\epsilon > 0$, conditional on the set-cover conjecture [15, Theorem 9]. Moreover, building upon exponential-time hypothesis, the prospect of obtaining an algorithm with runtime $2^{o(n)} \cdot \text{poly}(n, k)$ is unlikely [15, Theorem 6].

In the realm of fixed-parameter tractability (FPT), finding an optimal solution for k-median and k-means are known to be $\mathsf{W}[2]$-hard with respect to parameter k [13]. k-center and k-supplier are known to be $\mathsf{W}[1]$-hard with respect to parameter k [35, Theorem 1]. More recently, Cohen-Addad et al. [14] presented FPT approximation algorithms with respect to parameter k, with approximation ratio $1 + \frac{2}{e} + \epsilon$ and $1+\frac{8}{e}+\epsilon$ for k-median and k-means, respectively. Further, they showed that the approximation ratio is tight assuming gap exponential time hypothesis. Feldmann gave a factor 1.5 FPT approximation algorithm for k-center with respect to parameters k and h, where h is the highway dimension [35, Theorem 2]. Recently, Goyal and Jaiswal showed that a parameterized algorithm with respect to parameter k achieving approximation ratios $2 - \epsilon$ and $3 - \epsilon$ are unlikely for k-center and k-supplier, respectively, assuming $\mathsf{FPT} \neq \mathsf{W}[2]$ [19, Theorem 3].

Algorithmic fairness. The notion of fairness in algorithm design has gained significant traction in recent years. Many studies have considered cases where different social groups can be identified in datasets, based on ethnicity, socioeconomic background, gender, or other protected attributes. The output of an algorithm, while suitable when measured by a given objective function, might negatively impact one of said groups in a disproportionate manner [36]. In order to mitigate this shortcoming, constraints are imposed to promote more equitable outcomes. Various clustering formulations have been introduced, each catering to different fairness notions, which include: *proportionally fair*, ensuring that each cluster comprises of data points from every group [37]; *bounded representation*, guaranteeing that the representation of each group is bounded, preventing any single group from dominating a cluster [36, 38]; *fairness with outliers*, where the clustering solution may not be fair to a fraction of data points but pay a price for this deviation [39]; *cluster-center fairness*, ensuring a minimum (or maximum) number of cluster centers can be chosen from each group [20, 21]; and *socially fair*, aiming for equitable cost distribution among clusters [22, 40]. For a survey on fairness in clustering see Chhabra et al. [41] and tutorial resources by Brubach et al. [42].

Our emphasis in this work is on *cluster-center fairness*, which becomes particularly pronounced in data-summarization tasks, where the selected cluster centers serve as summary of overall data. Aiming to achieve such a notion of fairness, attention has turned to constraining the number of cluster centers to be chosen from each group. A notable problem variant is the *red-blue median problem* [20], in which the facilities are colored red or blue, and a solution may contain only up to a specified number of facilities (upper-bound) of each color [20]. This formulation was generalized by the *matroid-median problem* [24], where solutions must be independent sets of a matroid. Constant-factor approximation algorithms for both of these problems are known [20, 24, 43].

Kleindessner et al. [21] investigated fair-k-center, where a specific number of cluster centers must be chosen from each group. They propose a constant-factor pseudo-polynomial time algorithm, which was improved to a polynomial-time 3-approximation using maximal matching by Jones et al. [44]. In both studies, they focus on scenarios where the set of clients and facilities are identical, and the constraints on the number of cluster centers from each group are exact requirements (not lower-bounds), making the problem relatively easier. Chen et al. [45], extend the algorithm based on maximal matching to fair-k-supplier achieving a 5-approximation, where the set of clients and facilities can be different. In this paper, we improve the approximation ratio of fair-k-supplier to factor 3 and prove that the approximation ratio is tight.

In the above clustering formulations addressing cluster-center fairness, the constraints are either upper-bounds or exact requirements on the number of cluster centers from each group. Handling over-representation with upper-bound constraints, alone does not solve under-representation in clustering tasks, especially when the groups intersect. For an illustrative example, we refer to Thejaswi et al. [25, Figure 2] and discussion to follow.

Intersectionality and subgroup fairness. Hérbert-Johnson et al. [46] introduce the *multicalibration* framework, which aims to address calibration issues in machine-learning models related to intersectionality. Calibration, in this context, refers to the alignment of predicted probabilities with the actual frequencies of positive outcomes. *Multiaccuracy* introduced by Kim et al. [47] replaces calibration with accuracy constraints to propose a weaker fairness notion, which requires a predictor to be at least α-accurate for subgroups. Building upon these two notions, Gopalan et al. [48] introduce a hierarchy of weighted multicalibration, which finds a balance between multicalibration and multiaccuracy. These techniques rely on post-processing a machine-learning model and extensions to address in-processing-based fairness methods remains unclear.

Recently, Thejaswi et al. [25] introduced the diversity-aware k-median problem with *lower-bound* requirements on cluster centers from each group and overlapping facility groups. This increased level of generality increases the computational complexity of the problem, making it inapproximable in polynomial time and $\mathsf{W}[2]$-hard with respect to parameter k. In a subsequent work, Thejaswi et al. [26] present a parameterized approximation algorithm with respect to parameters k and t for diversity-aware k-median with $1 + \frac{2}{e} + \epsilon$ factor, where t is the number of groups. In this paper, we broaden our scope and introduce a generalized framework for solving diversity-aware clustering with k-median, k-means, and k-supplier objectives, accommodating overlapping facility groups.

Next, we formally introduce diversity-aware clustering problems and continue to study there computational complexity.

3 Diversity-aware clustering problems

We start our discussion by introducing the relevant terminology and the problem definitions.

In a k-clustering problem, we are given a metric space (U, d), a set of clients $C \subseteq U$, a set of facilities $F \subseteq U$, and an integer $k \in \mathbb{Z}_+$. The task is to find a subset $S \subseteq F$ of facilities of size k, which minimizes a certain clustering objective function $\phi(C, S)$. The clustering problems k-Median, k-Means, and k-Supplier are defined when the clustering objectives are $\sum_{c \in C} d(c, S)$, $\sum_{c \in C} d(c, S)^2$, and $\max_{c \in C} d(c, S)$, respectively. Here, $d(c, S)$ denotes the minimum distance between c and S, *i.e.*, $d(c, S) = \min_{s \in S} d(c, s)$.

For the diversity-aware clustering problem, we additionally consider a collection of possibly intersecting facility groups $\mathscr{G} = \{G_i\}_{i \in [t]}$ for some $t \in \mathbb{Z}_+$, where each subset $G_i \subseteq F$ corresponds to a facility group. We introduce a vector $\vec{r} = (r_i)_{i \in [t]}$, where each element r_i is associated with a group G_i and signifies a *lower-bound requirement* on the number of elements of G_i in a solution S. Thus, we aim to find a subset of facilities $S \subseteq F$ that satisfies three criteria: *(i)* $|S| = k$; *(ii)* S contains at least r_i facilities from each group G_i; and *(iii)* S optimizes a given clustering objective, which can be either k-Median, k-Means, or k-Supplier. We formally introduce the three variants of diversity-aware clustering problems below.

Problem 1 (Diversity-aware clustering) *Given a metric space (U, d), with clients $C \subseteq U$, facilities $F \subseteq U$, a collection $\mathscr{G} = \{G_i\}_{i \in [t]}$ of subsets of facilities $G_i \subseteq F$, a vector $\vec{r} = (r_i)_{i \in [t]}$ representing lower-bound requirements, and a non-negative integer k, find a subset $S \subseteq F$ of facilities of size $|S| = k$ satisfying the constraints $|S \cap G_i| \geq r_i$, for all $i \in [t]$, and the objective $\phi(C, S)$ is minimized. The diversity-aware k-median problem, the diversity-aware k-means problem, and the diversity-aware k-supplier problem are defined based on the formulation of the objective function $\phi(C, S)$, specifically, $\sum_{c \in C} d(c, S)$, $\sum_{c \in C} d(c, S)^2$, and $\max_{c \in C} d(c, S)$, respectively.*

Throughout this paper, a distinction is made based on whether the facility groups are mutually disjoint or possibly intersecting. As such, when the groups are disjoint we refer to our three clustering problems as *fair-k-median*, *fair-k-means*, and *fair-k-supplier*. Formally, we define fair clustering problems as follows.

Problem 2 (Fair clustering with disjoint facility groups) *Given metric space (U, d), with clients $C \subseteq U$, facilities $F \subseteq U$, a collection $\mathscr{G} = \{G_i\}_{i \in [t]}$ of disjoint subsets of facilities $G_i \subseteq F$, for some $t \in \mathbb{Z}_+$, and a vector $\vec{r} = (r_i)_{i \in [t]}$ of requirements, where $\sum_{i \in [t]} r_i = k$, find a subset $S \subseteq F$ of points such that exactly r_i points are chosen from each group G_i, and the clustering objective $\phi(C, S)$ is minimized. Based on the clustering objectives k-median, k-means, and k-supplier, the corresponding variants fair-k-median, fair-k-means, and fair-k-supplier are defined.*

In our method, we transform diversity-aware clustering and fair-clustering into instances of k-Median (k-Means and k-Supplier, respectively) with p-partition matroid constraints. This specific problem formulation is defined as follows:

Problem 3 (Clustering with p-partition matroid constraints) *Given a metric space (U, d), with a set $C \subseteq U$ of clients, a set $F \subseteq U$ of facilities, a collection $\mathscr{E} = \{E_1, \ldots, E_p\}$ of disjoint facility groups called a* p-partition matroid*, and an integer $k \in \mathbb{Z}_+$, find a subset of facilities $S \subseteq F$ of size $|S| = k$, containing at most one facility from each group E_i, and the clustering objective $\phi(C, S)$ is minimized. Based on the distinct clustering objectives, we introduce the variants of k-median, k-means, and k-supplier problems with p-partition matroid constraints.*

To isolate the problem from clustering objectives, we introduce *the diversity-requirements satisfiability problem*, which specifically aims to identify a subset of facilities $S \subseteq F$ of size $|S| = k$ that contain at least r_i facilities from each group G_i for every $i \in [t]$. This consideration is made disregarding the clustering objective $\phi(C, S)$. The formal definition of this problem is as follows.

Problem 4 (The diversity-requirements satisfiability problem) *Given a set of facilities F, a collection $\mathscr{G} = \{G_i\}_{i \in [t]}$ of facility subsets $G_i \subseteq F$ for all $i \in [t]$, a vector $\vec{r} = (r_i)_{i \in [t]}$ of lower-bound requirements and a non-negative integer $k \in \mathbb{Z}_+$, find a subset of facilities $S \subseteq F$ of size $|S| = k$ that satisfies the constraints $|S \cap G_i| \geq r_i$, for all $i \in [t]$.*

Throughout this manuscript, for brevity we refer to the diversity-aware clustering problems as Div-k-Median, Div-k-Means, and Div-k-Supplier; to the fair-clustering problems as Fair-k-Median, Fair-k-Means, and Fair-k-Supplier; and to clustering with p-partition matroid constraints as k-Median-p-PM, k-Means-p-PM, and k-Supplier-p-PM. Furthermore, we refer to the diversity-requirements satisfiability problem as Div-$\vec{r}$-Sat.

4 Computational complexity

In the next section, we discuss the computational complexity of diversity-aware clustering, particularly due to the intersecting facility groups. This comprehension is pivotal in appreciating the choice of algorithmic solutions proposed. Specifically, we establish the NP-hardness and W[1]-hardness with respect to various parameters. Further, we show inapproximability to any multiplicative factor even when exponential time with respect to various parameters is allowed.

4.1 Hardness and polynomial-time inapproximability

Diversity-aware clustering is an amalgamation of two independent problems. The first problem, referred as the requirement-satisfiability problem, seeks to determine a subset of facilities $S \subseteq F$ with cardinality $|S| = k$ that satisfies the constraints $|S \cap G_i| \geq r_i$, for all $i \in [t]$. The second problem pertains to minimizing the clustering objective $\phi(C, S)$. When the number of groups is $t = 1$, Div-k-Median, Div-k-Means, and Div-k-Supplier are equivalent to k-Median, k-Means, and k-Supplier, respectively. Therefore, the NP-hardness of the latter standard clustering problems implies the NP-hardness of their diversity-aware counterparts. Note that the NP-hardness refers to their respective decision variants.

Observation 4.1 *Problems* Div-k-Median*,* Div-k-Means*, and* Div-k-Supplier *are* NP*-hard.*

A stronger result can be established, namely, Div-k-Median, Div-k-Means, and Div-k-Supplier are NP-hard to approximate to any multiplicative factor in polynomial time. This inapproximability result can be derived from the fact that finding a feasible solution for Div-$\vec{r}$-Sat is NP-hard. The proof of Lemma 4.2 and Theorem 4.3 were originally presented by Thejaswi et al. [25]. Nevertheless, for the sake of completeness, we present these proofs in Appendix B.1 and B.2.

Lemma 4.2 *Problem* Div-$\vec{r}$-Sat *is* NP*-hard.*

The NP-hardness of Div-$\vec{r}$-Sat implies the polynomial-time inapproximability of Div-k-Median, Div-k-Means, and Div-k-Supplier to any multiplicative factor, as stated in the following theorem.

Theorem 4.3 *Assuming* P $\neq$ NP*, there exists no polynomial-time algorithm to approximate* Div-k-Median*,* Div-k-Means*, and* Div-k-Supplier *to any multiplicative factor.*

We continue to show that inapproximability results extend to specific cases with restricted input structures. The reduction in Theorem 4.3 confirms that Div-k-Median, Div-k-Means, and Div-k-Supplier remain inapproximable in polynomial-time to any multiplicative factor, even when every requirement is $\vec{r}$ is 1, that is $\vec{r} = \mathbf{1}_t$. Furthermore, Theorem 4.4 establishes polynomial-time inapproximability even when all subsets in collection $\mathscr{G} = \{G_i\}_{i\in[t]}$ have size precisely 2, i.e., $|G_i| = 2$, for all $i \in [t]$. Meanwhile, Theorem 4.5 indicates inapproximability when the underlying metric space is a tree metric.

Theorem 4.4 *Assuming* $\mathsf{P} \neq \mathsf{NP}$*, there exists no polynomial-time algorithm to approximate* DIV-k-MEDIAN*,* DIV-k-MEANS*, and* DIV-k-SUPPLIER *to any multiplicative factor, even if all the subsets in* $\mathscr{G} = \{G_i\}_{i\in[t]}$ *have size* $|G_i| = 2$*.*

Proof To establish this result, we reduce the vertex cover problem (VERTEXCOVER) to DIV-k-MEDIAN. The reductions for DIV-k-MEANS and DIV-k-SUPPLIER are analogous. In VERTEXCOVER, a graph $G = (V, E)$ and an integer k are given, and the objective is to decide if there exists a vertex subset $S \subseteq V$ of size $|S| = k$ such that for every edge $(u, v) \in E$ at least one of the vertices u or v is in S. The VERTEXCOVER problem is known to be NP-hard [49].

Consider an instance $(G = (V, E), k)$ of VERTEXCOVER with n vertices, m edges, and $k \leq |V|$. We construct an instance $((U, d), F, C, \mathscr{G}, \vec{r}, k)$ of DIV-k-MEDIAN as follows: First, we set $U = C = F = V$. Then, for each edge $\{u, v\} \in E$ we construct a group $G_i = \{u, v\}$, $\mathscr{G} = \{G_i\}_{i\in[m]}$. We set the lower-bound threshold $\vec{r} = \{1\}_{i\in[m]}$, and finally, the distance function is set to $d(u, v) = 1$, for all $(u, v) \in E$, and $d(u, v) = n+1$, for all $(u, v) \notin E$. The construction is polynomial in the size of the VERTEXCOVER instance.

Let $S \subseteq V$ be a solution to VERTEXCOVER. It is evident from the construction that $|S \cap G_i| \geq 1$ for each $G_i \in \mathscr{G}$ since each G_i represents a set of vertices in an edge. Additionally, as $|S| \leq k$, the set S also serves as a solution to DIV-k-MEDIAN. The argument in other direction follows a similar line of reasoning.

The construction demonstrates that if there exists a polynomial-time approximation algorithm for DIV-k-MEDIAN, we can solve VERTEXCOVER in polynomial time. However, this is unlikely under the premise that $\mathsf{P} \neq \mathsf{NP}$. The same argument applies for DIV-k-MEANS and DIV-k-SUPPLIER problems as well. ■

Theorem 4.5 *Assuming* $\mathsf{P} \neq \mathsf{NP}$*, there exists no polynomial-time algorithm to approximate* DIV-k-MEDIAN*,* DIV-k-MEANS*, and* DIV-k-SUPPLIER *to any multiplicative factor, even if the underlying metric space* (U, d) *is a tree metric.*

Proof We present a proof by contradiction. By the seminal work of Bartal [50], a metric space (u, d) can be embedded into a tree metric with at most $\log |U|$-factor distortion in distances. Consequently, DIV-k-MEDIAN, DIV-k-MEANS, and DIV-k-SUPPLIER can be embedded into a tree metric with at most $\log |U|$-factor distortion in distances. Assuming the existence of a polynomial-time α-approximation algorithm for DIV-k-MEDIAN on a tree metric, there exists a $\alpha \cdot \log |U|$-approximation algorithm for any metric instance of DIV-k-MEDIAN. A similar argument holds for DIV-k-MEANS and DIV-k-SUPPLIER. However, the existence of such an algorithm contradicts our inapproximability result in Theorem 4.3. Therefore, DIV-k-MEDIAN, DIV-k-MEANS, and DIV-k-SUPPLIER are NP-hard to approximate to any multiplicative factor, even if the underlying metric space is a tree. ■

The NP-hardness and inapproximability of diversity-aware clustering, even on simpler problem instances such as tree metrics with group size two, challenges our hope of designing polynomial-time algorithms even for structured inputs. It prompts us to explore super-polynomial time algorithms that remain efficient when certain parameters of the problem are small. Unfortunately, we show that this is not the case for Div-k-Median (Div-k-Means, Div-k-Supplier respectively) for several natural parameters of the problem.

4.2 Parameterised complexity of diversity-aware clustering

In essence, fixed parameter tractable (FPT) algorithms restricts the exponential dependence of running time to specific parameters. When these parameters are small, the runtime of the algorithm is also small. There exists a class of problems that are believed to be not fixed parameter tractable with respect to certain parameters of the problem, which collectively form the **W**-hierarchy. A brief introduction to parameterized complexity and related terminology is presented in Appendix A.

The reduction presented by Guha and Khuller [13, Theorem 3.1] implicitly establishes the W[2]-hardness of k-Median and k-Means with respect to parameter k. Similarly, k-Supplier is known to be W[2]-hard with respect to k as it is a generalization of k-DominatingSet, which also exhibits W[2]-hardness [51, Lemma 23.2.1]. When the number of groups $t = 1$, Div-k-Median, Div-k-Means, and Div-k-Supplier are equivalent to k-Median, k-Means, and k-Supplier, respectively. Consequently, the former problems are also W[2]-hard with respect to k. These findings are summarized in the following proposition.

Proposition 4.6 *Problems* Div-k-Median, Div-k-Means, *and* Div-k-Supplier *are* W[2]*-hard with respect to parameter k, which represents the size of the solution sought.*

To reinforce, by combining the reduction presented in Lemma 4.2 with the strong exponential time hypothesis (SETH) [52], we draw a compelling conclusion from the below corollary: focusing solely on the parameter k, the most viable approach for finding an optimal, or even an approximate solution for Div-k-Median, Div-k-Means, and Div-k-Supplier, is a straightforward exhaustive-search algorithm.

Corollary 4.7 *Assuming the strong exponential time hypothesis, for all $k \geq 3$ and $\epsilon > 0$, there exists no $\mathscr{O}(|F|^{k-\epsilon})$ algorithm to solve* Div-k-Median, Div-k-Means, *and* Div-k-Supplier *optimally. Furthermore, there exists no $\mathscr{O}(|F|^{k-\epsilon})$ algorithm to approximate* Div-k-Median, Div-k-Means, *and* Div-k-Supplier *to any multiplicative factor.*

Proof SETH implies that there is no $\mathscr{O}(|V|^{k-\epsilon})$ algorithm for k-DominatingSet, for any $\epsilon > 0$. The reduction in Lemma 4.2 creates an instance of Div-k-Median where the set of facilities $F = V$ and groups $\mathscr{G} = \{G_u = u \cup N(u)\}_{u \in V}$. Hence, any FPT exact or approximation algorithm running in time $\mathscr{O}(|F|^{k-\epsilon})$ for Div-k-Median can solve k-DominatingSet in $\mathscr{O}(|V|^{k-\epsilon})$ time, contradicting SETH. The same argument holds for Div-k-Means and Div-k-Supplier. ■

Given the W[2]-hardness of diversity-aware clustering problems with respect to k, it is natural to consider further relaxations on the running time of algorithms for the problem. An obvious question is whether we can approximate Div-k-Median, Div-k-Means, and Div-k-Supplier 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, Div-k-Means, and Div-k-Supplier captures k-DominatingSet from the reduction in Lemma 4.2, and finding a dominating set even of size $f(k)$ is $\mathsf{W}[1]$-hard [53].

Proposition 4.8 *For parameter k, problems* Div-k-Median, Div-k-Means, *and* Div-k-Supplier *are* $\mathsf{W}[1]$*-hard to approximate to any multiplicative factor, even when permitted to open $f(k)$ facilities, for any polynomial function f.*

Proof Any $\mathsf{FPT}(k)$ algorithm achieving a multiplicative factor approximation for Div-k-Median needs to solve Div-$\vec{r}$-Sat to satisfy requirements. Since these requirements capture k-DominatingSet Lemma 4.2, it means we can solve k-DominatingSet in $\mathsf{FPT}(k)$ time, which is a contradiction. Finally, noting the fact that finding dominating set of size $f(k)$, for any function f, is also $\mathsf{W}[1]$-hard due to [53, Theorem 1.3] concludes the proof. ■

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, k-Means, and k-Center are special case of Div-k-Median, Div-k-Means, and Div-k-Supplier, 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 $r = \max_{i\in[t]} r_i$ (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 $\mu \leq t$.

Proposition 4.9 *Problems* Div-k-Median, Div-k-Means, *and* Div-k-Supplier *are* $\mathsf{W}[2]$*-hard with respect to parameters $(k + t)$, $(r + t)$ and $(\mu + t)$.*

The above intractability results thwart our hopes of solving diversity-aware clustering either optimally or approximately. However, on a positive note, there exist instances where achieving a constant-factor approximation is possible in polynomial time. Next, we delve into these specific problem instances.

5 Polynomial-time approximable instances

Observe that, when the facility groups are disjoint, solving Div-$\vec{r}$-Sat is trivial as it involves choosing r_i facilities from each group G_i. Consequently, our inapproximability results do not hold under this condition. As highlighted by Thejaswi et al. [25][Section 4], certain instances of Div-k-Median that can be approximated to a constant factor. Specifically, when the facility groups are disjoint, Div-k-Median reduces to the matroid median problem (MatMedian), allowing the direct application of findings from MatMedian to achieve constant-factor approximation. The best known approximation for MatMedian stands at $7.081+\epsilon$ due to Krishnaswamy et al. [43][Section 6], which can be applied to Div-k-Median, thereby obtaining an algorithm with the same approximation ratio. In scenarios with two groups, a 5-approximation is possible using local search, as demonstrated by Thejaswi et al. [25][Section 4.2].

The matroid means problem (MatMeans) has the best known approximation of factor-64, as shown by Zhao et al. [54][Theorem 7]. Combining this result with the reduction outlined by Thejaswi et al. [25][Section 4.1] yields a factor-64 approximation algorithm for Div-k-Means with disjoint facility groups. Jones et al. [44] gave a 3-approximation algorithm for Fair-k-Center. Notably, this problem involves identical sets of facilities and clients, and picking exactly r_i centers from each group G_i, distinguishing it from our setting. Recently, Chen et al. [55] presented a polynomial time $(5+\epsilon)$-approximation algorithm for Div-k-Supplier, considering non-identical sets of clients and facilities with disjoint facility groups. In this work, we improve this approximation factor to $3+\epsilon$ and establish that the approximation ratio is tight under the assumption that $\mathsf{P} \neq \mathsf{NP}$.

In this work, we focus on the general variant of diversity-aware clustering with intersecting facility groups. As such, our algorithmic solutions are tailored to address the challenges posed by group intersections. Our complexity results have unequivocally established the infeasibility of achieving polynomial-time approximation, as outlined in Section 4. In the section to follow, we shift our attention to parameterized approximation algorithms for diversity-aware clustering.

6 Parameterized approximation algorithms

In this section, we introduce parameterized approximation algorithms for diversity-aware clustering. To maintain conciseness, we refer to a fixed parameter tractable algorithm as an FPT algorithm, with associated parameters specified in parentheses. For example, an FPT algorithm associated with parameter k is denoted as $\mathsf{FPT}(k)$. Throughout this section, our FPT algorithm pertains to parameters k, the size of solution sought and t, the number of groups, and it is denoted as $\mathsf{FPT}(k+t)$.

In this section, we present an $\mathsf{FPT}(k+t)$ algorithm for Div-k-Median, and further use these techniques to derive an algorithm for Div-k-Means. In addressing both Div-k-Median and Div-k-Means, our algorithm relies on constructing coresets of a size $\mathcal{O}(k \log |U| \nu^{-2})$, for some $\nu > 0$. However, techniques for constructing a coreset of similar size for Div-k-Supplier are presently not known. Hence, we present an alternative algorithmic approach to solve Div-k-Supplier in Section **??**.

6.1 Coresets for Div-k-Median and Div-k-Means

The high-level idea of coresets is to reduce the number of clients such that the distortion in the sum of distances is bounded by a multiplicative factor. Precisely, for every $\nu > 0$, we reduce the client set C to a weighted set C' with size $|C'| = \mathcal{O}(\Gamma k \log |U|)$, where the distortion in distances is bounded by a multiplicative factor $(1 \pm \nu)$ and $\Gamma > 0$. Formally, a coreset is defined as follows:

Definition 6.1 (Coreset) *Given an instance $((U,d), C, F, k)$ of k-Median (k-Means resp.) and a constant $\nu > 0$. A (strong)* coreset *is a subset $C' \subseteq C$ of clients 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:

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

To achieve this, we rely on the coreset construction by Cohen-Addad et al. [56].

Theorem 6.2 (Cohen-Addad et al. [56], Theorem 1) *Given a metric instance $((U,d),C,F,k)$ of k-Median or k-Means. For each $\nu > 0$, there exists an algorithm that, with probability at least $1-\delta$, computes a coreset $C' \subseteq C$ of size $|C'| = \mathscr{O}(\Gamma\, k \log|U|)$ in time $\mathscr{O}(|U|)$ such that $\Gamma = \min(\nu^{-2}+\nu^{-z}, k\nu^{-2})$, where $z = 1$ for k-Median and $z = 2$ for k-Means.*

To simplify subsequent discussions, we consider $\Gamma = 2\cdot\nu^{-2}$ and regard the coreset size as $\mathscr{O}(\nu^{-2}k\log|U|)$. Observe that the coresets obtained for k-Median and k-Means using the above theorem, also serve as coresets for k-Median-p-PM and k-Means-p-PM, respectively. Furthermore, the coreset construction remains consistent for Div-k-Median and Div-k-Means, given that their respective clustering objective remains unchanged with respect to k-Median and k-Means.

A birds-eye-view of our FPT algorithm is as follows: given a feasible instance of Div-k-Median, we start by enumerating collections of facility subsets that satisfy the lower-bound requirements. For each collection satisfying the criteria, we derive a constant-factor approximation of the optimal cost with respect to that particular collection. Given that at least one of these feasible solutions contain the optimal solution, the corresponding approximate solution obtained will serve as an approximation for Div-k-Median.

6.2 Enumerating feasible constraint patterns

We begin by defining the concept of a characteristic vector and a constraint pattern. Given an instance $((U,d),F,C,\mathscr{G},\vec{r},k)$ of diversity-aware clustering with $\mathscr{G} = \{G_i\}_{i\in[t]}$. A *characteristic vector* of a facility $f\in F$ is a vector denoted as $\vec{\chi}_f \in \{0,1\}^t$ such that i-th index of $\vec{\chi}_f$ is set to 1 if $f\in G_i$, 0 otherwise. Consider the set $\{\vec{\chi}_f\}_{f\in F}$ of the characteristic vectors of facilities in F. For each $\vec{\gamma}\in\{0,1\}^t$, let $E(\vec{\gamma}) = \{f\in F : \vec{\chi}_f = \vec{\gamma}\}$ denote the set of all facilities with characteristic vector $\vec{\gamma}$. Finally, $\mathscr{P} = \{E(\vec{\gamma})\}_{\vec{\gamma}\in\{0,1\}^t}$ induces a partition on F.

Definition 6.3 (Constraint pattern) *Given a k-multiset $\mathcal{E} = \{E(\vec{\gamma}_i)\}_{i\in[k]}$, where each $E(\vec{\gamma}_i)\in\mathscr{P}$, the* constraint pattern *associated with $\mathcal{E}$ is the vector obtained by the element-wise sum of the characteristic vectors $\{\vec{\gamma}_1,\ldots,\vec{\gamma}_k\}$, that is, $\sum_{i\in[k]}\vec{\gamma}_i$. A constraint pattern is said to be feasible if $\sum_{i\in[k]}\vec{\gamma}_i \ge \vec{r}$, where the inequality is taken element-wise.*

We can enumerate all feasible constraint patterns for Div-k-Median, Div-k-Means, or Div-k-Supplier in time $\mathscr{O}(2^{tk}t|U|)$, as asserted by the following lemma.

Lemma 6.4 *For a given instance of Div-k-Median, Div-k-Means, or Div-k-Supplier, we can enumerate all k-multisets with feasible constraint pattern in time $\mathscr{O}(2^{tk}t|U|)$.*

Proof First, we construct the set $\mathscr{P}$ in time $\mathscr{O}(2^t|F|)$, since $|\mathscr{P}| \leq 2^t$. Then, we enumerate all possible k-multisets of $\mathscr{P}$ that have feasible constraint pattern. Since, there are $\binom{|\mathscr{P}|+k-1}{k}$ possible k-multisets of $\mathscr{P}$, so enumerating all feasible constraint patterns can be done in $\mathscr{O}(|\mathscr{P}|^k t)$ time, resulting in total running time $\mathscr{O}(2^{tk}t|U|)$. ■

Facility groups with feasible constraint pattern $\mathcal{E} = \{E(\vec{\gamma})_i\}_{i\in[k]}$ might be duplicates. For instance, if it is necessary to choose two facilities from a group $E(\vec{\gamma})$, we make a copy of facilities in $E(\vec{\gamma})$ while introducing at most $\epsilon > 0$ distortion in distances, effectively forming a new group. Nonetheless, we consider every $\mathcal{E}$ with feasible constraint pattern to be disjoint.

Observe that for every k-multiset $\mathcal{E} = \{E(\vec{\gamma}_i)\}_{i\in[k]}$ with a feasible constraint pattern, choosing any arbitrary facility from each $E(\vec{\gamma}_i)$ yields a feasible solution to Div-$\vec{r}$-Sat. More specifically, every feasible constraint pattern induces a k-Median-k-PM, k-Means-k-PM or k-Supplier-k-PM instances, depending on the clustering objective.

6.3 FPT approximation algorithm for Div-k-Median and Div-k-Means

In this section we present one of our main results. First, we give an intuitive overview of our algorithm for Div-k-Median and Div-k-Means. As a warm-up, we describe a simple $(3+\epsilon)$-FPT-approximation algorithm. Further, show how to obtain a better guarantee, leveraging the recent FPT-approximation techniques of k-Median and k-Means.

Intuition:. Given an instance $((U,d), F, C, \mathscr{G}, \vec{r}, k)$ of Div-k-Median, we partition the facility set F into at most $\mathscr{O}(2^t)$ subsets $\mathscr{P} = \{E(\vec{\gamma})\}_{\vec{\gamma}\in\{0,1\}^t}$, such that each subset $E(\vec{\gamma})$ corresponds to the facilities with characteristic vector $\vec{\gamma} \in \{0,1\}^t$. Then, using Lemma 6.4, we enumerate all k-multisets of $\mathscr{P}$ with feasible constraint pattern. For each such k-multiset $\mathcal{E} = \{E(\vec{\gamma}_i)\}_{i\in[k]}$, we generate an instance $J_{\mathcal{E}} = ((U,d), \{E(\vec{\gamma}_i)\}_{i\in[k]}, C', k)$ of k-Med-k-PM, resulting in at most $\mathscr{O}(2^{tk})$ instances. Next, using Theorem 6.2, we build a coreset $C' \subseteq C$ of clients. Finally, we obtain an approximate solution to each k-Med-k-PM instance by adapting the techniques from Cohen-Addad et al. [14]. For Div-k-Means, the process is similar, except we create k-Means-k-PM instance using k-multiset $\mathcal{E} = \{E(\vec{\gamma}_i)\}_{i\in[k]}$ of feasible constraint pattern.

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

To find the closest client $\tilde{c}_i^*$ and its corresponding distance $\tilde{\lambda}_i^*$ in FPT time, we employ techniques of Cohen-Addad et al. [14], which they build on the work of Feldman and Langberg [57]. In our case, we use coreset construction by Cohen-Addad et al. [56].

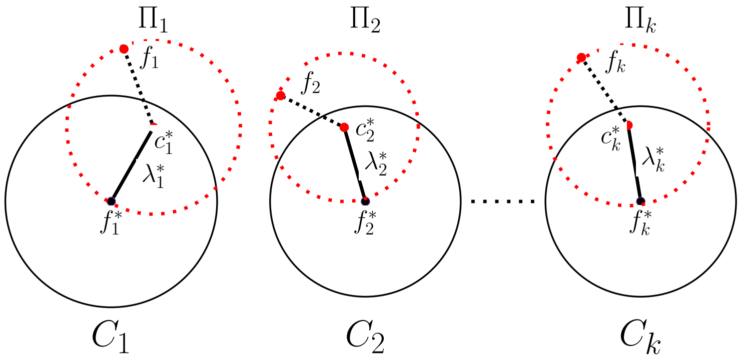

Fig. 1: An illustration of facility selection for the FPT algorithm for solving k-MED-k-PM instance.

The idea is to reduce the search spaces enough to allow brute-force search in FPT time. First, note that we already have a client coreset C' with $|C'| = O(k\nu^{-2}\log|U|)$ from Theorem 6.2. To find $\{\tilde{c}_i^*\}_{i\in[k]}$ we can enumerate all ordered k-multisets of C' in $\mathscr{O}((k\nu^{-2}\log|U|)^k)$ time. Then, to bound the search space of λ_i^* (which is at most $\Delta = \mathsf{poly}(|U|)$), we discretize the interval $[1, \Delta]$ to $[[\Delta]_\eta]$, for some $\eta > 0$. Note that $[\Delta]_\eta \leq \lceil \log_{1+\eta} \Delta \rceil = O(\log|U|)$. We thus enumerate all ordered k-multisets of $[[\Delta]_\eta]$ in at most $\mathscr{O}(\log^k|U|)$ time. The total time to find $\tilde{c}_i^*$ and $\tilde{\lambda}_i^*$ is thus $\mathscr{O}((k\nu^{-2}\log^2|U|)^k)$, which is $\mathsf{FPT}(k)$.

Next, using the facilities in $\{\widetilde{\Pi}_i^*\}_{i\in[k]}$, we find an approximate solution for the instance $J_{\tilde{\mathcal{E}}}$. As a warm-up, we show in Lemma 6.5 that picking exactly one facility from each $\widetilde{\Pi}_i^*$ arbitrarily already gives a $(3+\epsilon)$ approximate solution. This fact follows, since (U, d) is a metric space and satisfy triangle inequality. In Lemma 6.8 we obtain a better approximation ratio by modeling the k-MED-k-PM problem as a problem of maximizing a monotone submodular function, relying on the ideas of Cohen-Addad et al. [14].

Lemma 6.5 *For every $\epsilon > 0$, there exists a randomized $(3+\epsilon)$-approximation algorithm for* DIV-k-MEDIAN *which runs in time $f(k,t,\epsilon)\cdot \mathsf{poly}(|U|)$, where $f(k,t,\epsilon) = \mathscr{O}((2^t\epsilon^{-2}k^2\log k)^k)$.*

Proof Let $I = ((U,d), F, C, \mathscr{G}, \vec{r}, k)$ be an instance of DIV-k-MEDIAN. Let $J = ((U,d), C, \{E_1^*,\dots,E_k^*\}, k)$ be an instance of k-MED-k-PM corresponding to an optimal solution of I. That is, for some optimal solution $F^* = \{f_1^*,\dots,f_k^*\}$ of I, we have $f_j^* \in E_j^*$. Let $c_j^* \in C'$ be the closest client to f_j^*, for $j \in [k]$, with $d(f_j^*, c_j^*) = \lambda_j$. Now, consider the enumeration iteration where leader set is $\{c_j^*\}_{j\in[k]}$ and the radii is $\{\lambda_j^*\}$. The construction is illustrated in Figure 1.

We define Π_i^* to be the set of facilities in $E(\vec{\gamma}_i^*)$ at a distance of at most λ_i^* from c_i^*. We will now argue that picking one arbitrary facility from each Π_i^* gives a 3-approximation with respect to an optimal pick. Let $C_j^* \subseteq C'$ be a set of clients assigned to each facility f_j^* in optimal solution. Let $\{f_1,\dots,f_k\}$ be the arbitrarily chosen facilities, such that $f_j \in \Pi_j^*$. 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 $d(f_j^*, c_j^*) + d(c_j^*, f_j)) \leq 2\lambda_j^* \leq 2d(c, f_j^*)$, 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 [14], we obtain the approximation stated in the lemma. ■

Similarly for Div-k-Means, by considering squared distances, we obtain a $(9 + \epsilon)$-approximation algorithm with same time complexity.

Corollary 6.6 *For every $\epsilon > 0$, there exists a randomized $(9 + \epsilon)$-approximation algorithm for* Div-k-Means *which runs in time $f(k, t, \epsilon) \cdot \mathsf{poly}(|U|)$, where $f(k, t, \epsilon) = \mathcal{O}((2^t \epsilon^{-2} k^2 \log k)^k)$.*

Next, we focus on our main result, stated in Theorem 6.7. As mentioned before, we build upon the ideas for k-Median of Cohen-Addad et al. of [14]. Their algorithm, however, does not apply directly to our setting, as we have to ensure that the chosen facilities satisfy the constraints in $\vec{r}$. 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. The pseudocode of algorithms for Div-k-Median and k-Median-k-PM is available in Algorithm 1 and 2, respectively.

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 6.7.

Theorem 6.7 (Algorithm for Div-k-Median/Div-k-Means**)** *For every $\epsilon > 0$, there exists a randomized $(1 + \frac{2}{e} + \epsilon)$-approximation algorithm for* Div-k-Median *with running time $f(k, t, \epsilon) \cdot \mathsf{poly}(|U|, k, t)$, where $f(k, t, \epsilon) = \mathcal{O}\left(\left(\frac{2^t k^3 \log^2 k}{\epsilon^2 \log(1+\epsilon)}\right)^k\right)$. For* Div-$k$-Means*, with the same running time, we obtain a $(1 + \frac{8}{e} + \epsilon)$-approximation algorithm. Assuming* Gap-ETH*, the approximation ratios are tight for any* FPT$(k + t)$*-algorithm.*

Proof The hardness results follows from [14], which rules out factor $(1 + \frac{2}{e} - \epsilon)$-approximation algorithm for k-Median and $(1 + \frac{8}{e} - \epsilon)$-approximation algorithm for k-Means running in $\mathsf{FPT}(k)$ time, assuming Gap-ETH. Therefore, any approximation algorithm achieving aforementioned factors for Div-k-Median and Div-k-Means running in time $\mathsf{FPT}(k + t)$ also yields approximation algorithm for k-Median and k-Means with the same factors running in time $\mathsf{FPT}(k + 1) = \mathsf{FPT}(k)$, since $t = 1$ for these problems [1].

Our algorithm is described in detail in Algorithm 1. We primarily focus on Div-k-Median, indicating the parts of the proof for Div-k-Means. In essence, to

[1] In fact, [14] rules out much stronger running times. Further, using the same assumption, [58] strengthens the running time lower bound to rule out $g(k)|U|^{o(k)}$ time, for any function g, for k-Median and k-Means. This implies no $g(k + t)|U|^{o(k)}$ time approximation algorithm can achieve better factors than that of Theorem 6.7 for Div-k-Median and Div-k-Means. Note that, these problems can be exactly solved in time $|U|^{k+O(1)}$ time.

achieve the results for Div-k-Means, we need to consider squared distances which results in the claimed approximation ratio with same runtime bounds.

As mentioned before, to get a better approximation factor, the idea is to reduce the problem of finding an optimal solution to k-Median-k-PM to the problem of maximizing a monotone submodular function. To this end, for each $S \subseteq F$, we define the submodular function $\mathsf{improv}(S)$ that, in a way, captures the cost of selecting S as our solution. To define the function improv, we add a fictitious facility f'_i, for each $i \in [k]$ such that f'_i is at a distance $2\lambda^*_i$ for each facility in Π_i. We, then, use the triangle inequality to compute the distance of f'_i to all other nodes. Then, using an $(1-1/e)$-approximation algorithm (Line 12), we approximate improv. Finally, we return the set that has the minimum k-Median cost over all iterations.

Lemma 6.8 *Let $I = ((U,d), F, C, \mathscr{G}, \vec{r}, k)$ be an instance of* Div-k-Median *to Algorithm 1 and $F^* = \{f^*_i\}_{i\in[k]}$ being an optimal solution of I. Let $J = ((U,d), \{E^*_i\}_{i\in[k]}, C', k)$ be an instance of k-*Med-k-PM *corresponding to optimal solution F^*,* i.e., *$f^*_i \in E^*_i, i \in [k]$. On input (J, ϵ'), Algorithm 2 outputs a set $\hat{S}$ satisfying $\mathsf{cost}(\hat{S}) \leq (1 + \frac{2}{e} + \epsilon)\mathsf{cost}(F^*)$. Similarly, for* Div-$k$-Means, *$\mathsf{cost}(\hat{S}) \leq (1 + \frac{8}{e} + \epsilon)\mathsf{cost}(F^*)$.*

Proof [Sketch]

Algorithm 1: Div-k-Median$(I = ((U,d), F, C, \mathscr{G}, \vec{r}, k), \epsilon)$

Input: I, an instance of the Div-k-Median problem
ϵ, a real number
Output: T^*, subset of facilities

```
foreach γ⃗ ∈ {0,1}^t do
    E(γ⃗) ← {f ∈ F : γ⃗ = χ⃗_f}
𝓔 ← {E(γ⃗) : γ⃗ ∈ {0,1}^t}
C′ ← coreset((U,d), 𝓕, C, k, ν ← ε/16)

T* ← ∅
foreach multiset {E(γ⃗_1), ⋯ , E(γ⃗_k)} ⊆ 𝓔 of size k do
    if ∑_{i∈[k]} γ⃗_i ≥ r⃗, element-wise then
        Duplicate facilities to make subsets in {E(γ⃗_1), …, E(γ⃗_k)} disjoint
        T ← k-Median-PM((U,d), {E(γ⃗_1), ⋯ , E(γ⃗_k)}, C′, ε/4)
        if cost(C′, T) < cost(C′, T*) then
            T* ← T
return T*
```

Consider the iteration of Algorithm 2 where the chosen clients and radii are optimal, that is, $\lambda^*_i = d(c^*_i, f^*_i)$ 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,

• A: Are we planning to include the complete proofs in the appendix? If yes, we should indicate that the proofs are in the Appendix. Otherwise, we should indicate that they are present in the original article. Anyway, the current writing is confusing as it does not give a proof of the claimed factor. Maybe we should rename it to proof sketch. I already did it for Lemma 4.

Algorithm 2: k-Median-k-PM$(J = ((U,d), \{E_1, \cdots, E_k\}, C'), \epsilon')$

Input: J, an instance of the k-Median-k-PM problem
ϵ', a real number
Output: S^*, a subset of facilities

1 $S^* \leftarrow \emptyset$, $\eta \leftarrow e\epsilon'/2$
  foreach *ordered multiset* $\{c'_{i_1}, \cdots, c'_{i_k}\} \subseteq C'$ *of size* k **do**
2   **foreach** *ordered multiset* $\Lambda = \{\lambda_{i_1}, \cdots, \lambda_{i_k}\}$ *such that* $\lambda_{i_1} \subseteq [[\Delta]_\eta]$ **do**
3   **for** $j = 1$ *to* k **do**
4       $\Pi_j \leftarrow \{f \in E'_j \mid d_D(f, c'_{i_j}) = \lambda_{i_j}\}$
        Add a fictitious facility F'_j
        for $f \in \Pi_j$ **do**
5         $d(F'_j, f) \leftarrow 2\lambda_{i_j}$
6   **for** $f \notin \Pi_j$ **do**
7         $d(F'_j, v) \leftarrow 2\lambda_{i_j} + \min_{f \in \Pi_j} d(f, v)$
8   **for** $S \subseteq F$, *define* $\mathsf{improve}(S) := \mathsf{cost}(C', F') - \mathsf{cost}(C', F' \cup S)$ **do**
9       $S_{max} \leftarrow S \subseteq \mathcal{F}$ that maximizes $\mathsf{improve}(S)$ s.t. $|S \cap \Pi_j| = 1, \forall j \in [k]$
        if $\mathsf{cost}(C', S_{max}) < \mathsf{cost}(C', S^*)$ **then**
10        $S^* \leftarrow S$
11 **return** S^*

it is clear that in this iteration we have $f_i^* \in \Pi_i$ (see Algorithm 2, line 4). Furthermore, given the partition-matroid constraint imposed on it, the proposed submodular optimization scheme is guaranteed to pick exactly one facility from each of Π_i, for all i.

On the other hand, known results for submodular optimization show that this problem can be efficiently approximated within a factor $(1 - 1/e)$ of the optimum [59]. This translates to a $(1 + \frac{2}{e} + \epsilon)$-approximation $(1 + \frac{8}{e} + \epsilon$ resp.) of the optimal choice of facilities, one from each of Π_i [14]. (End of proof sketch of Lemma 6.8) ■

• A: Missing citations

(End of proof of Theorem 6.7) ■

Note that, Lemma 6.8 results in $(1 + \frac{2}{e} + \epsilon)$ and $(1 + \frac{8}{e} + \epsilon)$-approximation algorithms for Div-k-Median and Div-k-Means with disjoint facility groups, with running time $\mathcal{O}((k^2 \log^2 k \epsilon^{-2} \log(1+\epsilon)^{-1})^k \mathsf{poly}(|U|, k))$. Notably, it reduces the running time by a factor of 2^t as compared to corresponding problems with intersecting facility groups. Attaining this result involves transforming Div-k-Median with disjoint facility groups into an instance of k-Median-k-PM, by duplicating the facility group $G_i \in \mathcal{G}$ r_i times and some additional prepossessing. This process effectively generates dummy facility groups. We remark that Algorithm 2 can be extended to obtain similar results for k-Median-p-Partition and k-Means-p-Partition.

• A: I think it is not as straightforward as we claim here (transform lower bounds to upper bounds). It requires some preporssessing. We should explain that as well. We can do this in the journal version.

Corollary 6.9 (Algorithm for k-Median-k-PM/k-Means-k-PM) *For any* $\epsilon > 0$, *there exists a randomized* $(1 + \frac{2}{e} + \epsilon)$*-approximation algorithm for* k-Median-k-PM *with running time* $f(k,\epsilon)\cdot \mathsf{poly}(|U|,k)$*, where* $f(k,\epsilon) = \mathcal{O}\left(\left(\frac{k^2\log^2 k}{\epsilon^2\log(1+\epsilon)}\right)^k\right)$*. For* k-MEANS-k-PM*, with the same running time a* $(1+\frac{8}{e}+\epsilon)$*-approximation algorithm is achieved. Assuming* Gap-ETH*, the approximation ratios are tight for any* FPT(k)*-algorithm.*

6.4 FPT approximation algorithm for DIV-k-SUPPLIER

In our efforts to extend the coreset enumeration method described in the preceding subsection for addressing DIV-k-SUPPLIER, we faced challenges. The primary obstacle arose from the lack of a known methodology for constructing coresets of an appropriate size for k-SUPPLIER, say $f(k)\cdot\mathsf{poly}(\log n)$. Because of this limitation, we choose an alternative approach, circumventing coreset construction, and present a $(3+\epsilon)$-approximation algorithm for DIV-k-SUPPLIER with time complexity $\mathsf{FPT}(k+t)$.

An overview of our approximation algorithm for DIV-k-SUPPLIER is as follows: we employ a reduction technique, akin to previous sections, to transform an instance I of DIV-k-SUPPLIER into $\mathcal{O}(2^{tk})$ instances of k-SUPPLIER-k-PM in time $\mathcal{O}(2^{tk}k|U|)$. This reduction facilitates to identify instances whose feasible solutions satisfy the lower bound requirements of I. We guarantee that there is at least one feasible instance of k-SUPPLIER-k-PM whose optimum cost is at most the optimum cost of I. Building on the results of Chen et al.'s for FAIR-k-SUPPLIER, we can obtain a $(5+\epsilon)$-approximation for each k-SUPPLIER-k-PM instance in polynomial time [55]. By combining two steps yields a $(5+\epsilon)$-approximation algorithm for DIV-k-SUPPLIER with time $\mathsf{FPT}(k+t)$. However, we go a step further and present a 3-approximation for k-SUPPLIER-k-PM in polynomial time, thereby improving the approximation ratio Chen et al. [55]. Consequently, we obtain a $(3+\epsilon)$-approximation algorithm for DIV-k-SUPPLIER in $\mathsf{FPT}(k+t)$ time. The psuedocode is available in Algorithm 3. Remarkably, our 3-approximation algorithm for k-SUPPLIER-k-PM can be extended to obtain a 3-approximation for FAIR-k-SUPPLIER, the details are in Section 7.

First, we establish the following algorithmic guarantee for k-SUPPLIER-k-PM.

Theorem 6.10 (Algorithm for k-SUPPLIER-k-PM) *Algorithm 4 returns a 3-approximation for* k-SUPPLIER-k-PM *in time* $\mathcal{O}(k^{2+o(1)}|U|^2\log|U|)$*.*

Proof Let $J = ((U,d),F,\{E_1,\cdots,E_k\},C)$ be an instance of k-SUPPLIER-k-PM for Algorithm 4. Fix an optimal solution $O=\{o_1,\cdots,o_k\}$ to J. Let L^* be the cost of solution O. Note that $|O\cap E_i| = 1$ for every $i\in[k]$. So, without loss of generality, assume that facility $o_i \in E_i$.

Further, note that the cost of the optimal solution must be one of the $|U|^2$ entries corresponding to the distances between the pair of points in U (in fact, we can restrict to the distances in $F\cup C$). Hence, in the rest of the proof, we assume that we have guessed L^* correctly. Next lemma says that $X\subseteq C$, obtained after the while loop of Step 4 is actually a good approximation of O, albeit, X is infeasible since $X\cap E_i=\emptyset$, for $i\in[k]$.

Lemma 6.11 *For every* $c\in C$*, it holds that* $d(c,X)\le 2L^*$*. Moreover,* $|X|\le k$*.*

Proof First note that by construction every $c\in C$ is present in at least one $\mathsf{ball}(x,2L^*)$, $x\in X$ since the while loop of Step 4 terminates only when all the clients have been marked. For $c\in C$, we call $x_c\in X$ the *representative* of c if c was marked by Algorithm 4 because x_c was added to X. Hence, we have,

$$d(c, X) \leq d(c, x_c) \leq 2L^*.$$

Next, note that for $c, c' \in C$ belonging to the cluster of some $o_j \in O$, then $d(c, c') \leq d(c, o_j) + d(o_j, c') \leq 2L^*$. While, for every $x \neq x' \in X$, it holds that $d(x, x') > 2L^*$. Hence x and x' belong to different clusters of the optimal solution O. Therefore, we have that $|X| \leq k$. ■

The next steps of Algorithm 4, Step 7– 10, try to find a solution S^* using X that is feasible for J. Consider the graph $H = (X \cup [k], D)$ constructed by the algorithm. The following lemma is the key for the feasibility of S^*. It says that Step 8 of Algorithm 4 never fails.

Lemma 6.12 *There is matching in H on the vertices of X.*

Proof Since every $x_j \in X$ belongs to a different cluster in O, let $o_{x_j} \in O$ be the unique element of O such that x_j belongs to the cluster of o_{x_j}. Suppose $o_{x_j} \in E_{i_j}$. Then, we have $d(x_j, o_{x_j}) \leq L^*$ and hence there is an edge $(x_j, i_j) \in D$. Hence, $M = \{(x_j, i_j) \mid x_j \in X\}$ is matching on X. ■

Now, we are ready to show the claimed approximation factor of the theorem. Consider $c \in C$ and let $x_c \in X$ be its representative. Let $(x_c, i_c) \in M$ and let $f_{i_c} \in \mathsf{ball}(x_c, L^*) \cap E_{i_c}$ be the arbitrary facility added by the algorithm in Step 10 for (x_c, i_c). Hence, we have,

$$d(c, S^*) \leq d(c, x_c) + d(x_c, S^*) \leq d(c, x_c) + d(x_c, f_{i_c}) \leq 2L^* + L^* \leq 3L^*,$$

where the second last inequality follows from Lemma 6.11 and the fact that $f_{i_c} \in \mathsf{ball}(x_c, L^*)$.

Now it remains to show that S^* is a feasible solution, that is, $|S \cap E_i| = 1$ for every $i \in [k]$. Note that for $i \in [k]$ such that $(x, i) \in M$, since we add a facility from E_i to S^* in Step 10, we have $|S^* \cap E_i| = 1$. If there is an $i' \in [k]$ such that there is no $(x, i') \in M$, then we add an arbitrary facility from $E_{i'}$ to S^* in Step 12, and hence $|S^* \cap E_{i'}| = 1$. Thus, S^* is a feasible solution.

Running time. As mentioned previously, there are at most $|U|^2$ options for L^*, and therefore we can run the whole algorithm at most $|U|^2$ times looking for the smallest entry for which the algorithm is successful. However, we can find L^* faster by doing binary search in time $\mathscr{O}(\log |U|)$ for which we need to sort the entries requiring an additional time of $\mathscr{O}(|U|^2 \log |U|)$. Steps until (including) Step 6 can be performed in time $\mathscr{O}(|C|^2)$. Further, H can be constructed in time $\mathscr{O}(k^2|F|)$. The matching M of Step 8 can be found in time $\mathscr{O}(k^{2+o(1)})$ due to [60], since $|D| \leq k^2$. Finally, the remaining steps of Algorithm 4 can be done in time $\mathscr{O}(k|F|)$. Therefore, the overall running time of Algorithm 4 is $\mathscr{O}(|U|^2 \log |U| + \log |U|(|C|^2 + k^2|F| + k^{2+o(1)} + k|F|)) = \mathscr{O}(k^{2+o(1)}|U|^2 \log |U|)$. ■

Algorithm 3: Div-k-Supplier $(I = ((U, d), F, C, \mathscr{G}, \vec{r}, k))$

Input: I, an instance of the Div-k-Supplier problem
Output: T^*, subset of facilities that is a $(3 + \epsilon)$ approximation to I

foreach $\vec{\gamma} \in \{0, 1\}^t$ **do**
  $E(\vec{\gamma}) \leftarrow \{f \in F : \vec{\gamma} = \vec{\chi}_f\}$
$\mathscr{E} \leftarrow \{E(\vec{\gamma}) : \vec{\gamma} \in \{0, 1\}^t\}$
$T^* \leftarrow \emptyset$
foreach *multiset* $\{E(\vec{\gamma}_1), \cdots, E(\vec{\gamma}_k)\} \subseteq \mathscr{E}$ *of size* k **do**
  if $\sum_{i \in [k]} \vec{\gamma}_i \geq \vec{r}$, *element-wise* **then**
    Duplicate facilities to make subsets in $\{E(\vec{\gamma}_1), \ldots, E(\vec{\gamma}_k)\}$ disjoint
    $T \leftarrow k$-Center-PM$((U, d), \{E(\vec{\gamma}_1), \cdots, E(\vec{\gamma}_k)\}, C)$
    if $\mathit{cost}(C, T) < \mathit{cost}(C, T^*)$ **then**
      $T^* \leftarrow T$
return T^*

Algorithm 4: k-Supplier-k-PM $(J = ((U, d), F, \{E_1, \cdots, E_k\}, C))$

Input: J, an instance of the k-Supplier-k-PM problem
Output: S^*, subset of facilities that is a 3 approximation to J

$L^* \leftarrow$ guess the optimal cost of I
$X, S^* \leftarrow \emptyset$
Unmark all clients in C
while *there is an unmarked client* $c \in C$ **do**
  Add c to X
  Mark all the points in the $\mathsf{ball}(c, 2L^*)$
Let $H = (X \cup [k], D)$ be a bipartite graph such that $\forall (x, i) \in X \times [k], (x, i) \in D$ if $E_i \cap \mathsf{ball}(x, L^*) \neq \emptyset$
Let M be a matching in H on X
foreach $e = (x, i) \in M$ **do**
  add an arbitrary facility in $\mathsf{ball}(x, L^*) \cap E_i$ to S^*
while *there is an* $i \in [k]$ *that is unmatched in* M **do**
  add an arbitrary facility in E_i to S^*
return S^*

Our primary theoretical contribution for Div-k-Supplier is outlined in the following theorem.

Theorem 6.13 (Algorithm for Div-k-Supplier**)** *There exists a $(3 + \epsilon)$-approximation algorithm, for every $\epsilon > 0$, for* Div-k-Supplier *in time $f(k, t) \cdot \mathsf{poly}(|U|, t, k)$, where $f(k, t) = (2^{tk} k^{2+o(1)})$. Moreover, assuming $\mathsf{FPT} \neq W[2]$,* *there exists no FPT$(k+t)$ algorithm that computes $(3-\epsilon)$-approximation, for any $\epsilon > 0$, for* Div-k-Supplier.

• A: You should always start the statement with there exists algorithm that for every $\epsilon > 0$ works. IF you start otherwise, then there is an algorithm for every ϵ, which is insane. I had changed it before, but somehow it is turned into a wrong statement again!

Proof The pseudocode of this algorithm is presented in Algorithm 3, whose proof is outlined as follows: mirroring the approach as detailed in Section 6.2, we reduce Div-k-Supplier to $\mathcal{O}(2^{tk})$ instances of k-Supplier-k-PM. Let $I = ((U,d), F, C, \mathscr{G}, \vec{r}, k)$ be an instance of Div-k-Supplier, and let $J = ((U,d), F, \{E_1, \cdots, E_k\}, C))$ denote an instance of k-Supplier-k-PM corresponding to an optimal solution of I. Leveraging Theorem 6.10, we obtain a polynomial-time 3-approximation for J, which also serves as a $(3+\epsilon)$-approximation for I. By selecting a solution that minimizes the objective over all $\mathcal{O}(2^{tk})$ instances, we can derive a $(3+\epsilon)$-approximation algorithm for Div-k-Supplier.

The hardness result follows due to [61], who showed that k-Supplier is NP-hard to approximate within a factor better than 3 using a reduction from Hitting Set. Since Hitting Set is $W[2]$-hard, it implies that there is no FPT(k) algorithm that approximates k-Supplier better than factor 3. Further, since k-Supplier is equivalent to Div-k-Supplier when $t = 1$, the claimed hardness result follows[2].

■

7 Approximation algorithms for fair clustering

While addressing diversity-aware clustering problems, we introduced k-Median-k-PM, k-Means-k-PM, and k-Supplier-k-PM problems, where the facility groups form a partition. In these problems, the objective is to choose a subset of data points of size k, such that at most one facility should be chosen from each disjoint group, while simultaneously minimizing the clustering objective. Depending on the context, the clustering objective can be either k-median, k-means, or k-supplier. An important observation we make is that several fair clustering problems can be effectively reduced to the problem of clustering with partition matroid constraints. Through such reductions, our focus shifts from addressing the more complex problem of fair clustering formulations to the potentially simpler problems of k-Median-k-PM, k-Means-k-PM, and k-Supplier-k-PM.

In what follows, we establish a polynomial-time transformation from Fair-k-Median to k-Median-k-PM and demonstrate that an α-approximation for k-Median-k-PM allows us to derive a $\alpha(1+\epsilon)$-approximation algorithm for Fair-k-Median for any $\epsilon > 0$. We remark that our approach generalizes to Fair-k-Means and Fair-k-Supplier, but for brevity, we omit their detailed proofs.

Theorem 7.1 *Suppose there is an α-approximation algorithm for k-*Median*-k-PM, for $\alpha \geq 1$, in time $T(|F|, |C|, k)$. Then, there exists an $\alpha(1+\epsilon)$-approximation algorithm for* Fair-k-Median*, for every $\epsilon > 0$, in time $k|F| + T(k|F|, |C|, k)$.*

[2] In fact, assuming Gap-ETH, for any function g, [19] rules $g(k)|U|^{o(k)}$ time approximation algorithm for k-Supplier that achieves factor better than 3. This implies that assuming Gap-ETH, no approximation algorithm running in time $g(t+k)|U|^{o(k)}$ can achieve factor better than 3 for Div-k-Supplier. While Div-k-Supplier can be exactly solved in time $|U|^{k+O(1)}$.

Proof We sketch the proof. Consider an instance $I = (U, C, F, \mathscr{G} = \{G_i\}_{i\in[t]}, \vec{r}, k)$ of FAIR-k-MEDIAN, where we are required to select exactly r_i facilities from group $G_i \in \mathscr{G}$. Given $\epsilon > 0$, we construct an instance $J = (U', C, F', \{E_1, \cdots, E_k\}, k)$ of k-MEDIAN-k-PM as follows. For every $i \in [t]$ do the following. We create r_i copies of G_i in J and call them $E_i^1, \cdots, E_i^{r_i}$. For $f \in G_i$, let $f^1 \in E_i^1, \cdots, f^{r_i} \in E_i^{r_i}$ be its corresponding copies in J. Then, we set the distance between $f^j, f^{j'}$, for $j \neq j' \in [r_i]$, as $d(f^j, f^{j'}) = \frac{\epsilon}{9} \cdot d_{min}$, where $d_{min} := \min_{u\neq u'\in U} d(u, u')$, the minimum distance between a pair of points in U. This arrangement allows us to treat facilities f^j and $f^{j'}$ as distinct. Since $\sum_{i\in[t]} r_i = k$ in I, we have that there are exactly k partitions of F' in J. Note that $|F'| \leq k \cdot |F|$, implying that a k-MEDIAN-k-PM instance can have at most $|U'| \leq k \cdot |U|$ data points. This transformation can be done in time $k \cdot |F|$.

For the correctness, consider a feasible solution $X = \{x_1, \cdots, x_k\}$ of I, and let its cost be denoted as $\mathsf{cost}(X, I)$. Further, let $\Pi_X = \{\Pi_{x_1}, \cdots, \Pi_{x)k}\}$ be the clusters of P induced by X. Let $X'\{x'_1, \cdots, x'_k\}$ be a feasible solution of J obtained from X by taking the corresponding feasible copies of points of X. Then,

$$\mathsf{cost}(X', J) \leq \sum_{x'_i\in X'} \sum_{c\in\Pi_{x_i}} d(x'_i, c) \leq \sum_{x'_i\in X'} \sum_{c\in\Pi_{x_i}} d(x_i, c) + \frac{\epsilon}{9} \cdot d_{min} \leq (1 + \epsilon/9) \cdot \mathsf{cost}(X, I).$$

A similar argument shows that $\mathsf{cost}(X, I) \leq (1 + \frac{\epsilon}{9})\mathsf{cost}(X', J)$. Now consider an α-approximate solution A of J, and let B be the feasible solution for I that takes corresponding copies of points of X. Further, let O_I and O_J be optimal solutions for I and J, respectively.

$$\begin{aligned}
\mathsf{cost}(B, I) &\leq (1 + \epsilon/9)\, \mathsf{cost}(A, J) \\
&\leq \alpha \cdot (1 + \epsilon/9)\, \mathsf{cost}(O_J, J) \\
&\leq \alpha \cdot (1 + \epsilon/9)^3\, \mathsf{cost}(O_I, I) \\
&\leq \alpha \cdot (1 + \epsilon)\, \mathsf{cost}(O_I, I),
\end{aligned}$$

where the third inequality follows from second inequality by applying the above bounds on the cost twice. ∎

The techniques from the proof of the above theorem can be used to obtain similar results for the other problems.

Corollary 7.2 *Suppose there is an α-approximation algorithm for k-MEANS-k-PM, for $\alpha \geq 1$, in time $T(|F|, |C|, k)$. Then, there exists an $\alpha(1 + \epsilon)$-approximation algorithm for FAIR-k-MEANS, for every $\epsilon > 0$, in time $k|F| + T(k|F|, |C|, k)$.*

Corollary 7.3 *Suppose there is an α-approximation algorithm for k-SUPPLIER-k-PM, for $\alpha \geq 1$, in time $T(|F|, |C|, k)$. Then, there exists an $\alpha(1 + \epsilon)$-approximation algorithm for FAIR-k-SUPPLIER, for every $\epsilon > 0$, in time $k|F| + T(k|F|, |C|, k)$.*

Using the above results, we present approximation algorithms for FAIR-k-MEDIAN, FAIR-k-MEANS and FAIR-k-SUPPLIER.

Proposition 7.4 (Algorithm for FAIR-k-MEDIAN/FAIR-k-MEANS) *For every $\epsilon > 0$, there exists a $(1+\frac{2}{e}+\epsilon)$ and $(1+\frac{8}{e}+\epsilon)$ approximation algorithms for* FAIR-k-MEDIAN *and* FAIR-k-MEANS, *respectively, in time* $f(k,\epsilon) \cdot \mathsf{poly}(|U|,t,k)$*, where* $f(k,\epsilon) = \mathcal{O}\left(\left(\frac{k^2 \log^2 k}{\epsilon^2 \log(1+\epsilon)}\right)^k\right)$.

Proof From Theorem 6.7, we can derive $(1+\frac{2}{e}+\epsilon')$ and $(1+\frac{8}{e}+\epsilon')$ approximation algorithms for k-MEDIAN-k-PM and k-MEANS-k-PM, respectively, for some suitable $\epsilon' > 0$. Combining the outcomes of Theorem 6.7 with Theorem 7.1 and Corollary 7.2 finishes the proof. Note that the running time is contingent on the size of the coreset, and the number of duplicated facilities does not asymptotically increase the running time. ∎

Proposition 7.5 (Algorithm for FAIR-k-SUPPLIER) *For every* $\epsilon > 0$*, there exists a* $(3+\epsilon)$*-approximation algorithm for* FAIR-k-SUPPLIER *with time* $\mathcal{O}(k^{2+o(1)}|U|^2 \log|U|)$.

Proof From Theorem 6.10, a 3-approximation for k-SUPPLIER-k-PM is derived. Utilizing the construction outlined in Theorem 7.1, we can reduce a FAIR-k-SUPPLIER instance to a k-SUPPLIER-k-PM instance. By combining the outcomes of Theorem 6.10 and Corollary 7.3 we obtain a $3+\epsilon$ approximation for FAIR-k-SUPPLIER. The number of data points in k-SUPPLIER-k-PM instance can be at most $|U'| \leq k \cdot |U|$, resulting in the claimed running time. ∎

In fact, we get the following cleaner result for FAIR-k-SUPPLIER by modifying Algorithm 4. We omit the details.

Proposition 7.6 (Algorithm for FAIR-k-SUPPLIER) *There exists a 3-approximation algorithm for* FAIR-k-SUPPLIER *with time* $\mathcal{O}(k^{2+o(1)}|U|^2 \log|U|)$.

We remark that the approximation algorithm for FAIR-k-SUPPLIER runs in polynomial time, whereas, the approximation algorithms for FAIR-k-MEDIAN and FAIR-k-MEANS run in $\mathsf{FPT}(k)$ time.

Hardness of approximation. The tightness of the approximation ratios is established by considering that k-MEDIAN, k-MEANS, and k-SUPPLIER are instances of FAIR-k-MEDIAN, FAIR-k-MEANS, and FAIR-k-SUPPLIER, respectively, when the fair clustering variants have exactly one group. Consequently, the lower bound of the approximation ratio for the former problems is applicable to the respective latter problems. Specifically, the known NP-hardness result for approximating k-SUPPLIER to a factor of $3-\epsilon$, for any $\epsilon > 0$ [12][Theorem 6], extends to FAIR-k-SUPPLIER. Similarly, it is established that k-MEDIAN and k-MEANS cannot be approximated to factors of $(1+\frac{2}{e}-\epsilon)$ and $(1+\frac{8}{e}-\epsilon)$, respectively, for any $\epsilon > 0$, in time $\mathsf{FPT}(k) \cdot \mathsf{poly}(|U|,k)$ [14][Theorem 2]. These hardness results consequently apply to FAIR-k-MEDIAN and FAIR-k-MEANS, respectively.

• A: I have fixed the hardness results for diversity aware versions. Can you copy those statements here by removing t from them?

Observation 7.7 *Assuming* Gap-ETH*, for any* $\epsilon > 0$*, any* $(1+\frac{2}{e}-\epsilon)$*-approximation algorithm for* FAIR-k-MEDIAN *and any* $(1+\frac{8}{e}-\epsilon)$*-approximation for* FAIR-k-MEANS*, must run in time* $\mathcal{O}(|U|^{k^{g(\epsilon)}})$.

Observation 7.8 *Assuming* $\mathsf{P} \neq \mathsf{NP}$*, for any* $\epsilon > 0$*, it is* NP*-hard to approximate* FAIR-k-SUPPLIER *to factor* $3-\epsilon$.

• Aris: Need some conclusions.

Supplementary information.

Appendix A Parameterized complexity theory

In this section we define notation and terminology related to parameterized complexity. For terms and notations not defined here we refer the reader to a book by Cygan et al. [62].

Definition A.1 (Fixed-parameter tractability) *A problem P specified by input x and a parameter k is* fixed-parameter tractable *(FPT) if there exists an algorithm A to solve every instance $(x, k) \in P$ with running time of the form $f(k)|x|^{\mathcal{O}(1)}$, where $f(k)$ is function depending solely on the parameter k and $|x|^{\mathcal{O}(1)} = \mathrm{poly}(|x|)$ is a polynomial independent of the parameter k.*[3] *A problem P is* fixed-parameter intractable *otherwise if no algorithm with running time of the form $f(k)|x|^{\mathcal{O}(1)}$ exists to solve P.*

There exists a family of optimization problems for which no polynomial-time approximation algorithms are known, a general belief is that these problems are *inapproximable* in polynomial-time. However, we can try to design approximation algorithms for these problem in FPT time, which gives rise to *fixed-parameter approximation* algorithms, which we formally define now.

Definition A.2 (Fixed-parameter approximation) *An optimization problem P where each input instance $x \in P$ is associated with a corresponding optimal solution s^* and parameters $\{k_1, \dots, k_q\}$ is* fixed-parameter approximable *with respect to parameters $k_1, \dots, k_q$ if there exists an algorithm $\mathscr{A}$ that solves every instance $(x, k_1, \dots, k_q) \in P$ with running time of the form $f(k_1, \dots, k_q)|x|^{\mathcal{O}(1)}$ and returns a solution s with cost at most $\Phi(s) \leq \alpha \cdot \Phi(s^*)$, $\alpha \geq 1$ for a minimization problem, and at least $\Phi(s) \geq \alpha \cdot \Phi(s^*)$, $\alpha \leq 1$ for a maximization problem. Otherwise, P is* fixed-parameter inapproximable *with respect to parameters $k_1, \dots, k_q$.*

The complexity class FPT consists of problems for which parameterized algorithms are known. However, there are certain problems for which no known parameterized algorithms exist for various natural parameters of the problem, leading to the W-hierarchy. The W-hierarchy extends beyond FPT and comprises a hierarchy of complexity classes, *i.e.*,

$$\mathsf{FPT} \subseteq W[1] \subseteq W[2] \subseteq \dots$$

To establish that a problem belongs to a class in the W-hierarchy, it is essential to show a parameterized reduction from a known problem in the corresponding complexity class of the W-hierarchy. A parameterized reduction is slightly different than a many-to-one reduction used to establish NP-hardness and it is defined as follows.

Definition A.3 (Parameterized reduction) *Let P and P' be two problems. A* parameterized reduction *from P to P' is an algorithm $\mathscr{A}$ that transforms an instance $(x, k) \in P$ to an instance $(x', k') \in P'$ such that: (i) (x, k) is a* yes *instance of P if and only if (x', k') is a* yes *instance of P'; (ii) $k' \leq g(k)$ for some computable function g; and (iii) the running time of the transformation $\mathscr{A}$ is $f(k)|x|^{\mathcal{O}(1)}$, for some computable function f. Note that f and g need not be polynomial functions.*

For more details on parameterized complexity, we refer the reader to the book by Downey and Fellows [51], as well as the book by Cygan et al. [62].

[3] with an exception that, $\log n$ is allowed as part of function $f(k)$ if $n \ll k$.

Appendix B Proofs omitted in the main text

B.1 Proof of Lemma 4.2

Here, we present the proof establishing the NP-hardness of DIV-$\vec{r}$-SAT.

Proof We construct an instance of DIV-$\vec{r}$-SAT, denoted as $((U, d), F, \mathscr{G}, \vec{r}, k)$, from a given instance $(G = (V, E), k)$ of the dominating set problem (k-DOMINATINGSET). Note that, in the dominating set problem, a graph $G = (V, E)$ and an integer k are given, the objective is to decide if there exists a subset $S \subseteq V$ of vertices of cardinality $|S| = k$ such that, for every vertex $u \in V$, either u is in S or at least one neighbor of u in S, that is, $u \in S$ or $N(u) \cap S \neq \emptyset$.[4]

For an instance $(G = (V, E), k)$ of k-DOMINATINGSET, the construction of DIV-$\vec{r}$-SAT instance is as follows: set $U = F = V$, define the distance function $d(u, v) = 1$ for all $(u, v) \in U \times U$ and $u \neq v$. The facility groups $\mathscr{G} = \{G_u\}_{u \in V}$ are defined as $G_u = \{u\} \cup N(u)$, and the lower-bound thresholds are set as $\vec{r} = \mathbf{1}_{|V|}$, where $r_u = 1$ for all $u \in V$. The construction is polynomial in the size of the input.[5] Suppose $S \subseteq F$ is a feasible solution for DIV-$\vec{r}$-SAT. It's evident from the construction that S is a dominating set because $|G_u \cap S| \geq 1$, indicating that S intersects $\{u\} \cup N(u)$ for all $u \in V$. If $S \subseteq F$ is a feasible solution for k-DOMINATINGSET, then $S \cap G_u \neq \emptyset$ due to the fact that S is a dominating set and $G_u = u \cup N(u)$. This completes our proof. ■

B.2 Proof of Theorem 4.3

Here, we present the polynomial-time inapproximability results of DIV-k-MEDIAN, DIV-k-MEANS and DIV-k-CENTER.

Proof Assuming the existence of a polynomial-time algorithm that can approximate any feasible instance of DIV-k-MEDIAN (DIV-k-MEANS, DIV-k-CENTER, resp.) to a constant factor, we can use it to solve k-DOMINATINGSET in polynomial time. This contradiction is obtained through the reduction described in Lemma **??**. The approximate solution for DIV-k-MEDIAN (DIV-k-MEANS, DIV-k-CENTER, resp.) is also a valid solution for DIV-$\vec{r}$-SAT, and consequently for k-DOMINATINGSET, proving the polynomial-time inapproximability of DIV-k-MEDIAN (DIV-k-MEANS, DIV-k-CENTER resp.). ■

B.3 proof of Lemma 6.5

Proof Let $I = ((U, d), F, C, \mathscr{G}, \vec{r}, k)$ be an instance of DIV-k-MEDIAN. Let $J = ((U, d), C, \{E_1^*, \ldots, E_k^*\}, k)$ be an instance of k-MED-k-PM corresponding to an optimal solution of I. That is, for some optimal solution $F^* = \{f_1^*, \ldots, f_k^*\}$ of I, we have $f_j^* \in E_j^*$. Let $c_j^* \in C'$ be the closest client to f_j^*, for $j \in [k]$, with $d(f_j^*, c_j^*) = \lambda_j$. Now, consider the enumeration iteration where leader set is $\{c_j^*\}_{j \in [k]}$ and the radii is $\{\lambda_j^*\}$. The construction is illustrated in Figure 1.

[4] The neighbors of vertex $u \in V$ is denoted as $N(u)$, that is, $N(u) = \{v : (u, v) \in E\}$.

[5] A vector of length ℓ with all entries equal to 1 is denoted as $\mathbf{1}_\ell$.

We define Π_i^* to be the set of facilities in $E(\vec{\gamma}_i^*)$ at a distance of at most λ_i^* from c_i^*. We will now argue that picking one arbitrary facility from each Π_i^* gives a 3-approximation with respect to an optimal pick. 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 $f_j \in \Pi_j^*$. 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 $d(f_j^*, c_j^*) + d(c_j^*, f_j)) \leq 2\lambda_j^* \leq 2d(c, f_j^*)$, 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 [14], we obtain the approximation stated in the lemma. ■

B.4 Running time analysis for Theorem 6.7

First we bound the running time of Algorithm 2. Note that, the runtime of Algorithm 2 is dominated by the two *foreach* loops (Line 1 and 6), since the remaining steps, including finding an approximate solution to the submodular function improv, run in time $\mathsf{poly}(|U|)$. 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 $\mathcal{O}(([\Delta]_\eta)^k) = \mathcal{O}(\log_{1+\eta}^k |U|)$, since $\Delta = \mathsf{poly}(|U|)$. Hence, setting $\eta = \Theta(\epsilon)$, the overall running time of Algorithm 2 is bounded by[6]

$$\mathcal{O}\left(\left(\frac{k \log^2 |U|}{\epsilon^2 \cdot \log(1+\epsilon)}\right)^k \mathsf{poly}(|U|)\right) = \mathcal{O}\left(\left(\frac{k^3 \log^2 k}{\epsilon^2 \log(1+\epsilon)}\right)^k \mathsf{poly}(|U|)\right)$$

Since Algorithm 1 invokes Algorithm 2 $\mathcal{O}(2^{tk})$ times, its running time is bounded by $\mathcal{O}\left(\left(\frac{2^t k^3 \log^2 k}{\epsilon^2 \log(1+\epsilon)}\right)^k \mathsf{poly}(|U|)\right)$.

• Aris: Some reference contain ???'s I was not able to figure out why.

References

[1] Healey, J.F., Stepnick, A.: Diversity and Society: Race, Ethnicity, and Gender. Sage Publications, ??? (2019)

[2] Shore, L.M., Cleveland, J.N., Sanchez, D.: Inclusive workplaces: A review and model. Human Resource Management Review **28**(2), 176–189 (2018)

[3] Zinn, M.B., Dill, B.T.: Theorizing difference from multiracial feminism. Feminist Studies **22**(2), 321–331 (1996) https://doi.org/10.2307/3178416

[6] We use the fact that, if $k \leq \frac{\log |U|}{\log \log |U|}$, then $(k \log^2 |U|)^k = \mathcal{O}(k^k \mathsf{poly}(n))$, otherwise if $k \geq \frac{\log |U|}{\log \log |U|}$, then $(k \log^2 n)^k = \mathcal{O}(k^k (k \log k)^{2k})$.

[4] Crenshaw, K.: Demarginalizing the intersection of race and sex: A black feminist critique of antidiscrimination doctrine, feminist theory and antiracist politics. In: Feminist Legal Theories, pp. 23–51. Routledge, ??? (2013)

[5] Runyan, A.S.: What is intersectionality and why is it important? Academe **104**(6), 10–14 (2018)

[6] Fish, S.: Reverse racism, or how the pot got to call the kettle black. Atlantic Monthly **272**(5), 128–136 (1993)

[7] Kearns, M., Neel, S., Roth, A., Wu, Z.S.: An empirical study of rich subgroup fairness for machine learning. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 100–109 (2019)

[8] Kasy, M., Abebe, R.: Fairness, equality, and power in algorithmic decision-making. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, pp. 576–586 (2021)

[9] Ghosh, A., Genuit, L., Reagan, M.: Characterizing intersectional group fairness with worst-case comparisons. In: Artificial Intelligence Diversity, Belonging, Equity, and Inclusion, pp. 22–34 (2021). PMLR

[10] Hoffmann, A.L.: Where fairness fails: data, algorithms, and the limits of antidiscrimination discourse. Information, Communication & Society **22**(7), 900–915 (2019)

[11] Kong, Y.: Are "intersectionally fair" ai algorithms really fair to women of color? a philosophical analysis. In: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency. FAccT '22, pp. 485–494. ACM, ??? (2022)

[12] Hochbaum, D.S., Shmoys, D.B.: A unified approach to approximation algorithms for bottleneck problems. Journal of the ACM (JACM) **33**(3), 533–550 (1986)

[13] Guha, S., Khuller, S.: Greedy strikes back: Improved facility location algorithms. In: SODA, pp. 649–657. SIAM, USA (1998)

[14] Cohen-Addad, V., Gupta, A., Kumar, A., Lee, E., Li, J.: Tight fpt approximations for k-median and k-means. In: Proceedings of the International Colloquium on Automata, Languages, and Programming (2019). Dagstuhl

[15] Fomin, F.V., Golovach, P.A., Inamdar, T., Purohit, N., Saurabh, S.: Exact Exponential Algorithms for Clustering Problems. In: International Symposium on Parameterized and Exact Computation. LIPIcs, vol. 249, pp. 13–11314. Dagstuhl, ??? (2022)

[16] Charikar, M., Guha, S., Tardos, E., Shmoys, D.B.: A constant-factor approximation algorithm for the k-median problem (extended abstract). In: Proceedings of

the Thirty-First Annual ACM Symposium on Theory of Computing, pp. 1–10. ACM, ??? (1999)

[17] Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., Pandit, V.: Local search heuristic for k-median and facility location problems. In: Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing, pp. 21–29 (2001)

[18] Li, S.: Approximating capacitated k-median with $(1 + \epsilon)k$ open facilities. In: Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 786–796 (2016). SIAM

[19] Goyal, D., Jaiswal, R.: Tight fpt approximation for socially fair clustering. Information Processing Letters **182**, 106383 (2023)

[20] Hajiaghayi, M., Khandekar, R., Kortsarz, G.: Local search algorithms for the red-blue median problem. Algorithmica **63**, 795–814 (2012)

[21] Kleindessner, M., Awasthi, P., Morgenstern, J.: Fair k-center clustering for data summarization. In: ICML, pp. 3448–3457 (2019). PMLR

[22] Ghadiri, M., Samadi, S., Vempala, S.: Socially fair k-means clustering. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, pp. 438–448 (2021)

[23] Chen, D.Z., Li, J., Liang, H., Wang, H.: Matroid and knapsack center problems. Algorithmica **75**(1), 27–52 (2016)

[24] Krishnaswamy, R., Kumar, A., Nagarajan, V., Sabharwal, Y., Saha, B.: The matroid median problem. In: SODA (2011)

[25] Thejaswi, S., Ordozgoiti, B., Gionis, A.: Diversity-aware k-median: clustering with fair-center representation. In: Machine Learning and Knowledge Discovery in Databases. Research Track, pp. 765–780 (2021). Springer

[26] Thejaswi, S., Gadekar, A., Ordozgoiti, B., Osadnik, M.: Clustering with fair-center representation: parameterized approximation algorithms and heuristics. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1749–1759 (2022)

[27] Charikar, M., Guha, S., Tardos, Shmoys, D.B.: A constant-factor approximation algorithm for the k-median problem. JCSS **65**(1), 129–149 (2002)

[28] Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., Pandit, V.: Local search heuristics for k-median and facility location problems. SIAM Journal on computing **33**(3), 544–562 (2004)

[29] Cohen-Addad, V., Gupta, A., Hu, L., Oh, H., Saulpic, D.: An improved local

search algorithm for k-median. In: Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1556–1612 (2022). SIAM

[30] Byrka, J., Pensyl, T., Rybicki, B., Srinivasan, A., Trinh, K.: An improved approximation for k-median, and positive correlation in budgeted optimization. In: SODA, pp. 737–756 (2014). SIAM

[31] Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: A local search approximation algorithm for k-means clustering. Computational Geometry **28**(2-3), 89–112 (2004)

[32] Ahmadian, S., Norouzi-Fard, A., Svensson, O., Ward, J.: Better guarantees for k-means and euclidean k-median by primal-dual algorithms. SIAM Journal on Computing **49**(4), 17–97 (2019)

[33] Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. Theoretical computer science **38**, 293–306 (1985)

[34] Hsu, W.-L., Nemhauser, G.L.: Easy and hard bottleneck location problems. Discrete Applied Mathematics **1**(3), 209–215 (1979)

[35] Feldmann, A.E., Marx, D.: The parameterized hardness of the k-center problem in transportation networks. Algorithmica **82**, 1989–2005 (2020)

[36] Chierichetti, F., Kumar, R., Lattanzi, S., Vassilvitskii, S.: Fair clustering through fairlets. Advances in neural information processing systems **30** (2017)

[37] Chen, X., Fain, B., Lyu, L., Munagala, K.: Proportionally fair clustering. In: International Conference on Machine Learning, pp. 1032–1041 (2019). PMLR

[38] Ahmadian, S., Epasto, A., Kumar, R., Mahdian, M.: Fair correlation clustering. In: International Conference on Artificial Intelligence and Statistics, pp. 4195–4205 (2020). PMLR

[39] Bandyapadhyay, S., Inamdar, T., Pai, S., Varadarajan, K.: A constant approximation for colorful k-center. In: 27th Annual European Symposium on Algorithms (ESA 2019), vol. 144, pp. 12–11214. Dagstuhl, Dagstuhl, Germany (2019)

[40] Abbasi, M., Bhaskara, A., Venkatasubramanian, S.: Fair clustering via equitable group representations. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, pp. 504–514 (2021)

[41] Chhabra, A., Masalkovaitė, K., Mohapatra, P.: An overview of fairness in clustering. IEEE Access **9**, 130698–130720 (2021)

[42] Brubach, B., Chakrabarty, D., Dickerson, J.P., Seyed Esmaeili, M.K., Knittel, M., Morgenstern, J., Samadi, S., Srinivasan, A., Tsepenekas, L.: Fairness in clustering. https://www.fairclustering.com/ (2022)

[43] Krishnaswamy, R., Li, S., Sandeep, S.: Constant approximation for k-median and k-means with outliers via iterative rounding. In: STOC, pp. 646–659 (2018)

[44] Jones, M., Nguyen, H., Nguyen, T.: Fair k-centers via maximum matching. In: ICML, pp. 4940–4949 (2020). PMLR

[45] Chen, X., Ji, S., Wu, C., Xu, Y., Yang, Y.: An approximation algorithm for diversity-aware fair k-supplier problem. Theoretical Computer Science **983**, 114305 (2024)

[46] Hébert-Johnson, U., Kim, M., Reingold, O., Rothblum, G.: Multicalibration: Calibration for the (computationally-identifiable) masses. In: International Conference on Machine Learning, pp. 1939–1948 (2018). PMLR

[47] Kim, M., Reingold, O., Rothblum, G.: Fairness through computationally-bounded awareness. Advances in neural information processing systems **31** (2018)

[48] Gopalan, P., Kim, M.P., Singhal, M.A., Zhao, S.: Low-degree multicalibration. In: Conference on Learning Theory, pp. 3193–3234 (2022). PMLR

[49] Garey, M.R., Johnson, D.S.: Computers and Intractability vol. 29. W. H. Freeman and Co., ??? (2002)

[50] Bartal, Y.: Probabilistic approximation of metric spaces and its algorithmic applications. In: FOCS, pp. 184–193 (1996). IEEE

[51] Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity vol. 4. Springer, ??? (2013)

[52] Impagliazzo, R., Paturi, R.: On the complexity of k-sat. JCSS **62**(2), 367–375 (2001)

[53] S., K.C., Laekhanukit, B., Manurangsi, P.: On the parameterized complexity of approximating dominating set. Journal of ACM **66**(5) (2019)

[54] Zhao, A., Zhou, Y., Liu, Q.: Improved approximation algorithms for matroid and knapsack means problems. International Journal of Foundations of Computer Science, 1–21 (2023)

[55] Chen, X., Ji, S., Wu, C., Xu, Y., Yang, Y.: An approximation algorithm for diversity-aware fair k-supplier problem. Theoretical Computer Science, 114305 (2023)

[56] Cohen-Addad, V., Saulpic, D., Schwiegelshohn, C.: A new coreset framework for clustering. In: Proceedings of the ACM SIGACT Symposium on Theory of Computing, pp. 169–182 (2021)

[57] Feldman, D., Langberg, M.: A unified framework for approximating and clustering

data. In: STOC, pp. 569–578. ACM, ??? (2011)

[58] Manurangsi, P.: Tight running time lower bounds for strong inapproximability of maximum k-coverage, unique set cover and related problems (via t-wise agreement testing theorem). In: Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '20, pp. 62–81. Society for Industrial and Applied Mathematics, USA (2020)

[59] Calinescu, G., Chekuri, C., Pal, M., Vondrák, J.: Maximizing a monotone submodular function subject to a matroid constraint. SIAM Journal on Computing **40**(6), 1740–1766 (2011)

[60] Chen, L., Kyng, R., Liu, Y.P., Peng, R., Gutenberg, M.P., Sachdeva, S.: Maximum flow and minimum-cost flow in almost-linear time. In: 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS), pp. 612–623 (2022). https://doi.org/10.1109/FOCS54457.2022.00064

[61] Hochbaum, D.S., Shmoys, D.B.: A unified approach to approximation algorithms for bottleneck problems. Journal of ACM **33**(3), 533–550 (1986) https://doi.org/10.1145/5925.5933

[62] Cygan, M., Fomin, F.V., Kowalik, Ł., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, ??? (2015)