On Coresets for Fair Clustering in Metric and Euclidean Spaces and Their Applications

Sayan Bandyapadhyay
Department of Informatics, University of Bergen, Norway

Fedor V. Fomin
Department of Informatics, University of Bergen, Norway

Kirill Simonov
Department of Informatics, University of Bergen, Norway

Abstract

Fair clustering is a variant of constrained clustering where the goal is to partition a set of colored points. The fraction of points of each color in every cluster should be more or less equal to the fraction of points of this color in the dataset. This variant was recently introduced by Chierichetti et al. [NeurIPS 2017] and became widely popular. This paper proposes a new construction of coresets for fair k-means and k-median clustering for Euclidean and general metrics based on random sampling. For the Euclidean space $\mathbb{R}^d$, we provide the first coresets whose size does not depend exponentially on the dimension d. The question of whether such constructions exist was asked by Schmidt, Schwiegelshohn, and Sohler [WAOA 2019] and Huang, Jiang, and Vishnoi [NeurIPS 2019]. For general metric, our construction provides the first coreset for fair k-means and k-median.

New coresets appear to be a handy tool for designing better approximation and streaming algorithms for fair and other constrained clustering variants. In particular, we obtain

- the first fixed-parameter tractable (FPT) PTAS for fair k-means and k-median clustering in $\mathbb{R}^d$. The near-linear time of our PTAS improves over the previous scheme of Böhm, Fazzone, Leonardi, and Schwiegelshohn [ArXiv 2020] with running time $n^{\text{poly}(k/\epsilon)}$;
- FPT "true" constant-approximation for metric fair clustering. All previous algorithms for fair k-means and k-median in general metric are bicriteria and violate the fairness constraints;
- FPT 3-approximation for lower-bounded k-median improving the best-known 3.736 factor of Bera, Chakrabarty, and Negahbani [ArXiv 2019];
- the first FPT constant-approximations for metric chromatic clustering and ℓ-Diversity clustering;
- near linear-time (in n) PTAS for capacitated and lower-bounded clustering improving over PTAS of Bhattacharya, Jaiswal, and Kumar [TOCS 2018] with super-quadratic running time;
- a streaming $(1+\epsilon)$-approximation for fair k-means and k-median of space complexity polynomial in k, d, ϵ and $\log n$ (the previous algorithms have exponential space complexity on either d or k).

2012 ACM Subject Classification Theory of computation → Facility location and clustering

Keywords and phrases fair clustering, coresets, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.23

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2007.10137

Funding This work is supported by the Research Council of Norway via the project "MULTIVAL".

Acknowledgements The authors are thankful to Vincent Cohen-Addad for sharing the full version of [19].

© Sayan Bandyapadhyay, Fedor V. Fomin, and Kirill Simonov;
licensed under Creative Commons License CC-BY 4.0

48th International Colloquium on Automata, Languages, and Programming (ICALP 2021).
Editors: Nikhil Bansal, Emanuela Merelli, and James Worrell; Article No. 23; pp. 23:1–23:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Given a set of n data points in a metric space and an integer k, *clustering* is the task of partitioning the points into k groups or clusters so that the points in each cluster are similar. In this paper, we consider clustering problems with *fairness* constraints. While there are many competing notions of fairness in the literature, here we consider clustering with fairness constraints or fair clustering as introduced by Chierichetti et al. [16] in their seminal work. The notion became widely popular within a short period triggering a large body of new work [38, 7, 9, 28, 4, 11, 15, 1, 32]. The idea of fair clustering is to enforce additional (fairness) constraints to remove the inherent bias or discrimination from vanilla (unconstrained) clustering. For example, suppose we have a sensitive feature (e.g, race or gender). We want to find a clustering where the fraction of points from a traditionally underrepresented group in every cluster is more or less equal to the fraction of points from this group in the dataset. Indeed, the work of Chierichetti et al. [16] shows that clustering computed by classical vanilla algorithms can lead to widely varied ratios for a particular group, especially when the number of clusters is large enough.

In this work, we consider the fair clustering model independently formulated by Bercea et al. [9] and Bera et al. [7]. In this model, we are given ℓ groups $P_1, \ldots, P_\ell$ of points in a metric space and balance parameters $\alpha_i, \beta_i \in [0, 1]$ for each group $1 \leq i \leq \ell$. A clustering is *fair* if the fraction of points from group i in every cluster is at least β_i and at most α_i. Additionally, in [7], the groups are allowed to overlap, i.e, a point can belong to multiple protected classes. We refer to the fair clustering problem with overlapping groups as (α, β)*-fair clustering*. We note that this is the most general version of fair clustering considered in the literature, and this is the notion of fairness we adapt in this paper. Both [9] and [7] obtain polynomial time $O(1)$-approximation for this problem that violates the fairness constraints by at most small additive factors.

We denote by Γ the maximum number of distinct collections of groups to which a point belongs. If all the groups are disjoint, then $\Gamma = \ell$. Note that if every point belongs to at most Λ groups, then Γ is at most ℓ^{Λ}. As noted in [7, 28], while Λ can very well be more than 1, it is usually a constant in most of the applications. Thus, in this case, $\Gamma = \ell^{O(1)}$, which is expected to be much smaller compared to n, the total number of points in the union of the groups.

Several works related to fair clustering were devoted to scalability [28, 38, 11, 4]. Along this line, in a beautiful work, Schmidt et al. [38] defined coresets for fair clustering. Note that a coreset for a center-based vanilla clustering problem is roughly a summary of the data that for every set C of k centers approximately (within $(1 \pm \epsilon)$ factor) preserves the optimal clustering cost. Over the years, researchers have paid increasing attention to the design of coreset construction algorithms to optimize the coreset size. Indeed, finding smaller coresets continues to be an active research area in the context of vanilla k-median and k-means clustering. For general metric spaces, the best-known upper bound on coreset size is $O((k \log n)/\epsilon^2)$ [21] and the lower bound is known to be $\Omega((k \log n)/\epsilon)$ [5]. For Euclidean spaces of dimension d, it is possible to construct coresets, based on random sampling, of size $(k/\epsilon)^{O(1)}$ [22, 39, 29, 13], which in particular does not depend on n and d.

In the vanilla version of a clustering problem, given the cluster centers, clusters are formed by assigning each point to its nearest center. In contrast, in a constrained version, such an assignment might not lead to a clustering that satisfies the constraints. Hence, for fair clustering, we need a stronger definition of coreset. Accordingly, Schmidt et al. [38] initiated the study of fair coresets. Schmidt et al. [38] and subsequently Huang et al. [28]

designed deterministic algorithms in $\mathbb{R}^d$ that construct fair coresets whose sizes exponentially depend on d. To remove this exponential dependency on d, Schmidt et al. [38] proposed an interesting open question whether it is possible to use random sampling for construction of fair coresets. Huang et al. [28] also suggested the same open question. Besides, Huang et al. asked whether it is possible to achieve a similar size bound as in the vanilla setting.

1.1 Our Results and Contributions

We study fair clustering under the k-median and k-means objectives. Our first main result is the following theorem.

▶ **Theorem 1.** *There is an $O(n(k+\ell))$ time randomized algorithm that w.p. at least $1-1/n$, computes a coreset of size $O(\Gamma(k\log n)^2/\epsilon^3)$ for (α,β)-fair k-median and $O(\Gamma(k\log n)^7/\epsilon^5)$ for (α,β)-fair k-means, where Γ is the number of distinct collections of groups to which a point may belong. If the groups are disjoint, the algorithm runs in $O(nk)$ time. Moreover, in $\mathbb{R}^d$, the coreset sizes are $O\left(\frac{\Gamma}{\epsilon^3}\cdot k^2\log n(\log n + d\log(1/\epsilon))\right)$ for (α,β)-fair k-median*[1] *and $O\left(\frac{\Gamma}{\epsilon^5}\cdot k^7(\log n)^6(\log n + d\log(1/\epsilon))\right)$ for (α,β)-fair k-means.*

Theorem 1 provides the first coreset construction for fair clustering problem in general metric spaces. Note that if the number of groups is just 1, we obtain coresets of size $O_\epsilon(\mathrm{poly}(k\log n))$, which is somewhat comparable to the best-known bound of $O_\epsilon(k\log n)$ [21] in the vanilla case. We also note that this is the first sampling based coreset construction scheme for fair clustering, and in $\mathbb{R}^d$, the first coreset construction scheme where the size of the coreset does not depend exponentially on the dimension d (see Table 1). In fact, the dependency on d is only linear. Specifically, our result improves the bound (for k-median) in [28] by a factor of $\Theta\left(\frac{\epsilon^{-d+3}}{\log n(\log n+d)}\right)$ (see Table 1). Thus, if d is sufficiently large, our coreset size is much smaller compared to theirs. In fact, the dependency of the previous coreset size on n can be super-polylogarithmic (or super-polynomial) when d is large, whereas ours is only polylogarithmic in the worst case. In the light of the above discussion, our result solves the open question proposed in [38] and partly solves the open question proposed in [28].

Table 1 Previous and current coreset results in $\mathbb{R}^d$.

	k-median		k-means	
	size	construction time	size	construction time
[38]			$O(\Gamma k\epsilon^{-d-2}\log n)$	$O(k\epsilon^{-d-2}n\log n)$
[28]	$O(\Gamma k^2\epsilon^{-d})$	$O(k\epsilon^{-d+1}n)$	$O(\Gamma k^3\epsilon^{-d-1})$	$O(k\epsilon^{-d+1}n)$
This	$O(\frac{\Gamma}{\epsilon^3}\cdot k^2\log n(\log n+ d\log(1/\epsilon)))$	$O(nd(k+\ell))$	$O(\frac{\Gamma}{\epsilon^5}\cdot k^7(\log n)^6(\log n+ d\log(1/\epsilon)))$	$O(nd(k+\ell))$

Actually, our coreset construction scheme, similar to [38, 28], is much more general in the following sense. The coreset can preserve not only the cost of optimal fair clustering, but also the cost of any optimal clustering with group-cardinality constraints. In particular, for

[1] The Euclidean version is not a special case of the general metrics case, as here the set of potential centers is infinite.

any clustering problem with constraints that can be expressed in terms of the number of elements that go from each group to each cluster (formally defined in Section 2), we obtain a small size coreset. This gives rise to new coresets for a wide range of clustering problems including lower-bounded clustering [40, 2, 8].

We further exploit the new coreset construction to design clustering algorithms in various settings. In general metrics, we obtain the first fixed-parameter tractable (FPT) constant-factor approximation for (α, β)-fair clustering with parameters k and Γ. That is, the running time of our algorithm is exponential only in the values k and Γ while polynomial in the size of the input. All previous constant-approximation algorithms were bicriteria and violated the fairness constraints by some additive factors. Hence, the study of FPT approximation is well-motivated. Our approximation factors are reasonably small and improve the best-known approximation factors of the existing bicriteria algorithms (see Table 2). Moreover, our coreset leads to improved constant FPT approximations for many other clustering problems. For example, we obtain an improved ≈ 3-approximation algorithm for lower-bounded k-median [40, 2, 8] that is FPT parameterized by k. Previously, the best-known factor for FPT approximation for this problem was 3.736 [8].

Based on our coreset, we also obtain the first FPT $(1+\epsilon)$-approximation for (α, β)-fair clustering in $\mathbb{R}^d$ with parameters k and Γ. For constant Γ, the running time of our algorithm is near-linear and up to $\log n$, matches the time of the algorithm of Kumar, Sabharwal, and Sen for vanilla clustering [34]. A comparison with the running time of the previous $(1+\epsilon)$-approximation algorithms can be found in Table 3. We also obtain FPT $(1+\epsilon)$-approximations with parameter k for the Euclidean version of several other problems including capacitated clustering [19, 17] and lower-bounded clustering. We note that these are the first $(1+\epsilon)$-approximations for these problems with near-linear dependency on n. For Euclidean capacitated clustering, quadratic time FPT algorithms follow due to [20, 10] (see Table 4). Also, the $(1+\epsilon)$-approximation for Euclidean capacitated clustering in [19] and [17] have running time $(k\epsilon^{-1})^{k\epsilon^{-O(1)}} n^{O(1)}$ and at least $n^{\epsilon^{-O(1)}}$ (see Table 4).

Table 2 Approximation results for (α, β)-fair clustering in general metrics. "multi" denotes if the algorithm can handle overlapping groups. In "approx." columns, the first (resp. second) value in a tuple is the approximation factor (resp. violation). [7] does not explicitly compute the $O(1)$ factor, but it is $> 3+\epsilon$ (resp. $> 9+\epsilon$) for k-median (resp. k-means), where ϵ is a sufficiently large constant.

	multi	k-median		k-means	
		approx.	time	approx.	time
[9]		$(4.675, 1)$	poly(n)	$(62.856, 1)$	poly(n)
[7]	✓	$(O(1), 4\Lambda + 3)$	poly(n)	$(O(1), 4\Lambda + 3)$	poly(n)
This		≈ 3	$(k\ell)^{O(k\ell)} n \log n$	≈ 9	$(k\ell)^{O(k\ell)} n \log n$
This	✓	≈ 3	$(k\Gamma)^{O(k\Gamma)} n \log n$	≈ 9	$(k\Gamma)^{O(k\Gamma)} n \log n$

Our coreset also leads to small space $(1+\epsilon)$-approximation in streaming setting for (α, β)-fair clustering in $\mathbb{R}^d$ when the groups are disjoint. We show how to maintain an $O(d^2\ell \cdot \text{poly}(k \log n)/\epsilon^4)$ size coreset in each step. One can apply our $(1+\epsilon)$-approximation algorithm on the coreset to compute a near-optimal clustering. In the previous streaming algorithms [38], the space complexity depended exponentially on either d or k.

Our coreset construction scheme is based on the algorithm for the vanilla case due to Chen [14]. Chen showed that a small number of points can be chosen randomly from a metric space (in a clever way) that form a coreset for vanilla clustering. Our construction first divides the input points into a number of subsets, and for each subset, applies Chen's algorithm to compute a coreset for that subset. Our final coreset is the union of all these computed coresets. Surprisingly, our algorithm for fair clustering is as simple as that. Our main technical contribution is the analysis of this algorithm, which shows that the returned weighted subset of the points is a coreset for fair clustering with high probability.

Table 3 Running time of the $(1+\epsilon)$-approximations for fair clustering in $\mathbb{R}^d$.

	running time	version
[38]	$n^{O(k/\epsilon)}$	2-color, $(1,k)$-fair clustering
[11]	$n^{\text{poly}(k/\epsilon)}$	ℓ-color, $(1,k)$-fair clustering
This	$2^{\tilde{O}(k/\epsilon^{O(1)})}(k\Gamma)^{O(k\Gamma)}nd\log n$	(α,β)-fair clustering

Although our algorithm follows the framework of Chen's, the proof that the algorithm correctly computes a fair coreset is much more complicated. Our analysis is strongly inspired by the analysis of Cohen-Addad and Li [19] of Chen's algorithm in the context of capacitated clustering. However, there are two major difficulties of applying the analysis technique of [19] directly to our problem. Firstly, in our case input points belong to groups which can overlap, and secondly, fairness constraints are much more general compared to capacity constraints. The novelty of our work lies in overcoming the two above-mentioned hurdles. To overcome the first hurdle, we divide the input points into equivalence classes based on their group membership and using a probabilistic argument prove that it is possible to consider these classes separately. To overcome the second hurdle, we prove that fairness constraints can be treated as a collection of independent constraints each of which is similar to a capacity constraint. All these ideas together help us generalize the approach of [19] to fair clustering.

Table 4 Running time of the $(1+\epsilon)$-approximations for capacitated clustering in $\mathbb{R}^d$.

	running time
[20]	$2^{\text{poly}(k/\epsilon)}n^2(\log n)^{k+2}d$
[10]	$2^{\tilde{O}(k/\epsilon^{O(1)})}\cdot n^2(\log n)^2 d$
[19]	$(k\epsilon^{-1})^{k\epsilon^{-O(1)}}n^{O(1)}$
[17]	$n^{\epsilon^{-O(1)}}$ $(d=2)$ $n^{(\log n/\epsilon)^{O(d)}}$ $(d\geq 3)$
This	$2^{\tilde{O}(k/\epsilon^{O(1)})}nd^{O(1)}+nk^2\epsilon^{-O(1)}\log n$

Apart from the coreset construction, the novelty of our work lies in the design of an algorithm for computing the minimum cost fair assignment to given centers, based on mixed-integer linear programming. This is the heart of all our approximation algorithms.

1.2 Related Work

Schmidt et al. [38] defined the concept of fair coresets and gave coreset of size $O(\ell k \epsilon^{-d-2} \log n)$ for the disjoint group case of Euclidean (α, β)-fair k-means. They also gave an $n^{O(k/\epsilon)}$ time $(1+\epsilon)$-approximation for the two-color version of the problem. Using the framework in [25], Huang et al. [28] improved the coreset size bound of [38] by a factor of $\Theta\left(\frac{\log n}{\epsilon k^2}\right)$ and gave the first coreset for Euclidean (α, β)-fair k-median of size $O(\Gamma k^2 \epsilon^{-d})$. Böhm et al. [11] obtained near-linear time constant-approximation in a restricted setting. They also obtained an $n^{\text{poly}(k/\epsilon)}$ time $(1+\epsilon)$-approximation for the Euclidean version in the same setting.

Chierichetti et al. [16] gave a polynomial time $\Theta(t)$-approximation for a special version of (α, β)-fair k-median with two groups, where t is a balance parameter. Bera et al. [7] obtained polynomial time $O(1)$-approximation for (α, β)-fair clustering that violates the fairness constraints by at most an additive factor of $4\Lambda + 3$. For the disjoint group case, their violation factor is only 3. Independently, Bercea et al. [9] obtained algorithms with the same approximation guarantees as in [7] for the disjoint version, but with at most 1 additive factor violation. For other related works on fair clustering, see [4, 16, 37, 9, 7, 1, 15, 32, 33]. For related works on coresets, see [27, 26, 23, 35, 21, 22, 39, 6, 12, 29].

2 Preliminaries

In all the clustering problems we study in this paper, we are given a set P of points in a metric space $(\mathcal{X}, d(\cdot,\cdot))$, that we have to cluster. We are also given a set F of cluster centers in the same metric space. We note that P and F are not-necessarily disjoint, and in fact, P may be equal to F. We assume that the distance function $d(\cdot,\cdot)$ is provided by an oracle that for any given $x, y \in \mathcal{X}$ in constant time returns $d(x, y)$. In the Euclidean version of a clustering problem, $P \subseteq \mathbb{R}^d$, $F = \mathbb{R}^d$ and $d(\cdot,\cdot)$ is the Euclidean metric.[2] In the metric version, we assume that F is finite. Thus, strictly speaking, the Euclidean version is not a special case of the metric version. In the metric version, we denote $|P \cup F|$ by n and in the Euclidean version, $|P|$ by n. For any set S and a point p, $d(p, S) := \min_{q \in S} d(p, q)$. Also, for any integer $t \geq 1$, we denote the set $\{1, 2, \ldots, t\}$ by $[t]$.

In the k-median problem, given an additional parameter k, the goal is to select a set of at most k centers $C \subset F$ such that the quantity $\sum_{p \in P} d(p, C)$ is minimized. k-means is identical to k-median, except here we would like to minimize $\sum_{p \in P} (d(p, C))^2$.

Next, we define our notion of fair clustering following the definition in [7].

▶ **Definition 2** (Definition 1, [7]). *In the fair version of a clustering problem (k-median or k-means), one is additionally given ℓ many (not necessarily disjoint)* groups *of P, namely $P_1, P_2, \ldots, P_\ell$. One is also given two* fairness vectors *$\alpha, \beta \in [0,1]^\ell$, $\alpha = (\alpha_1, \ldots, \alpha_\ell)$, $\beta = (\beta_1, \ldots, \beta_\ell)$. The objective is to select a set of at most k centers $C \subset F$ and an assignment $\varphi : P \to C$ such that φ satisfies the following* fairness constraints*:*

$$|\{x \in P_i : \varphi(x) = c\}| \leq \alpha_i \cdot |\{x \in P : \varphi(x) = c\}|, \quad \forall c \in C, \forall i \in [\ell],$$
$$|\{x \in P_i : \varphi(x) = c\}| \geq \beta_i \cdot |\{x \in P : \varphi(x) = c\}|, \quad \forall c \in C, \forall i \in [\ell],$$

and $\text{cost}(\varphi)$ *is minimized among all such assignments.*

In the (α, β)-Fair k-median problem, $\text{cost}(\varphi) := \sum_{x \in P} d(x, \varphi(x))$, and in the (α, β)-Fair k-means problem, $\text{cost}(\varphi) := \sum_{x \in P} d(x, \varphi(x))^2$. To refer to these two problems together, we will use the term (α, β)-Fair Clustering. We call φ that satisfies the fairness constraints *a*

[2] Due to the lack of better notations, we denote the dimension by d and distance function by $d(\cdot,\cdot)$.

fair assignment. We denote the minimum cost of a fair assignment of a set of points P to a set of k centers C by faircost(P, C), and faircost(P) denotes the minimum of faircost(P, C') over all possible sets of k centers C'.

Next, we state our notion of coresets. We follow the definitions in [38, 28]. For a clustering problem with k centers and ℓ groups $P_1, \ldots, P_\ell$, a *coloring constraint* is a $k \times \ell$ matrix M having non-negative integer entries. The entry of M corresponding to row i and column j is denoted by M_{ij}. Next, we have the following observation, which was also noted in [38, 28].

▶ **Proposition 3.** *Given a set C of k centers, the assignment restriction required for (α, β)-Fair Clustering can be expressed as a collection of coloring constraints.*

In our definition, a coreset is required to preserve the optimal clustering cost w.r.t. all coloring constraints, and hence it also preserves the optimal fair clustering cost. Next, we formally define the cost of a clustering w.r.t. a set of centers and a coloring constraint.

First, consider the k-median objective. Suppose we are given a weight function $w : P \to \mathbb{R}_{\geq 0}$ (non-negative reals). Let $W \subseteq P \times \mathbb{R}$ be the set of pairs $\{(p, w(p)) \mid p \in P \text{ and } w(p) > 0\}$. For a set of centers $C = \{c_1, \ldots, c_k\}$ and a coloring constraint M, wcost(W, M, C) is the minimum value $\sum_{p \in P, c_i \in C} \psi(p, c_i) \cdot d(p, c_i)$ over all assignments $\psi : P \times C \to \mathbb{R}_{\geq 0}$ such that

1. For each $p \in P$, $\sum_{c_i \in C} \psi(p, c_i) = w(p)$.
2. For each $c_i \in C$ and group $1 \leq j \leq \ell$, $\sum_{p \in P_j} \psi(p, c_i) = M_{ij}$.

For k-means, wcost(W, M, C) is defined in the same way except it is the minimum value $\sum_{p \in P, c_i \in C} \psi(p, c_i) \cdot d(p, c_i)^2$. If there is no such assignment ψ, wcost$(W, M, C) = \infty$. When $w(p) = 1$ for all $p \in P$, we simply denote W by P and wcost(W, M, C) by cost(P, M, C). Now we define a coreset. We call it universal coreset, as it is required to preserve optimal clustering cost w.r.t. all coloring constraints.

▶ **Definition 4** (Universal coreset). *For a given unweighted point set P and a clustering objective, a universal coreset is a set of weighted points $W \subseteq P \times \mathbb{R}$ such that for every set of centers C of size k and any coloring constraint M,*

$$(1 - \epsilon) \cdot cost(P, M, C) \leq wcost(W, M, C) \leq (1 + \epsilon) \cdot cost(P, M, C).$$

3 Our Techniques

Here, we describe the techniques and key ideas used to obtain the new results of the paper. The detailed version of our results and formal proofs appear in the attached full version. For simplicity, we limit our discussion to k-median clustering. We start with the coreset results.

3.1 Universal Coreset Construction

Our coreset construction algorithms are based on random sampling and we will prove that our algorithms produce universal coresets with high probability (w.h.p.). At a first glance, it is not easy to see how to sample points in the overlapping group case, as the decision has an effect on multiple groups. For simplicity, first we discuss the disjoint group case.

The Disjoint Group Case

Our coreset construction algorithm is built upon the coreset construction algorithm for vanilla clustering due to Chen [14]. In our case, we have points from ℓ disjoint color classes. So, we apply Chen's algorithm for each color class independently. Note that Chen's algorithm was used to show that for any given set of centers C, the constructed coreset approximately preserves the optimal clustering cost. However, we would like to show that for any given set of centers C, the constructed coreset approximately preserves the optimal clustering cost corresponding to any given constraint M. At this stage, it is not clear why Chen's algorithm should work in such a generic setting. Our main technical contribution is to show that sampling based approaches like Chen's algorithm can be used even for such a stronger notion of universal coreset. We will try to give some intuition after describing our algorithm. Our algorithm is as follows.

Given the set of points P, first we apply the algorithm of Indyk [30] for computing a vanilla k-median clustering of P. This is a bicriteria approximation algorithm that uses $O(k)$ centers and runs in $O(nk)$ time. Let C^* be the set of computed centers, ν be the constant approximation factor and Π be the cost of the clustering. Also, let $\mu = \Pi/(\nu n)$ be a lower bound on the average cost of the points in any optimal k-median clustering. Note that for any point p, $d(p, C^*) \leq \Pi = \nu n \cdot \mu$.

For each center $c_i^* \in C^*$, let $P_i^* \subseteq P$ be the corresponding cluster of points assigned to c_i^*. We consider the ball $B_{i,j}$ centered at c_i^* and having radius $2^j\mu$ for $0 \leq j \leq N$, where $N = \lceil \log(\nu n) \rceil$. We note that any point at a distance $2^N\mu \geq \nu n \cdot \mu$ from c_i^* is in $B_{i,N}$, and thus all the points in P_i^* are also in $B_{i,N}$. Let $B'_{i,0} = B_{i,0}$ and $B'_{i,j} = B_{i,j} \setminus B_{i,j-1}$ for $1 \leq j \leq N$. We refer to each such $B'_{i,j}$ as a ring for $1 \leq i \leq k, 0 \leq j \leq N$. For each $0 \leq j \leq N$ and color $1 \leq t \leq \ell$, let $P'_{i,j,t}$ be the set of points in $B'_{i,j}$ of color t, and $P_{i,j} = \cup_{t=1}^{\ell} P'_{i,j,t}$. Let $s = \Theta(k \log n/\epsilon^3)$ for a sufficiently large constant hidden in $\Theta(.)$.

For each center $c_i^* \in C^*$, we perform the following steps.

Random Sampling. For each color $1 \leq t \leq \ell$ and ring index $0 \leq j \leq N$, do the following. If $|P'_{i,j,t}| \leq s$, add all the points of $P'_{i,j,t}$ to $W_{i,j}$ and set the weight of each such point to 1. Otherwise, select s points from $P'_{i,j,t}$ independently and randomly (without replacement) and add them to $W_{i,j}$. Set the weight of each such point to $|P'_{i,j,t}|/s$.

The set $W = \cup_{i,j} W_{i,j}$ is the desired universal coreset. As the number of rings is $O(k \log n)$, the size of W is $O(\ell(k \log n)^2/\epsilon^3)$. From [14], it follows that for each color, the coreset points can be computed in time linear in the number of points of that color times $O(k)$. Thus, our coreset construction algorithm runs in $O(nk)$ time. Next, we show that W is indeed a universal coreset w.h.p.

Note that we need to show that for any set of centers C, the optimal clustering cost is approximately preserved w.r.t. all possible combinations of cluster sizes as defined by the constraint matrices. In Chen's analysis, it was sufficient to argue that for any set of centers C, the optimal clustering cost needs to be preserved. This seems much easier compared to our case. (Obviously, the details are much more complicated even in the vanilla case.) For example, in the vanilla case, let $p \in P$ be a point that is assigned to a center $c \in C$ in an optimal clustering. Note that c must be a closest center to p. For simplicity, suppose p has a unique closest center. Now, if p is chosen in the coreset, then the total weight of p must also be assigned to c in any optimal assignment w.r.t. C. Thus, the assignment function for original and coreset points remains same in the vanilla case. This fact is in the heart of their analysis. Note that this is not necessarily true in our case. We cannot just use the nearest neighbor assignment scheme, as in our case cluster sizes are predefined through M. Indeed, in our case we might very well need to assign the weight of a coreset point to multiple centers to satisfy M. In general, this is the main hurdle one faces while analyzing a sampling based approach for fair coreset construction.

For analyzing our algorithm, we follow an approach similar to the one by Cohen-Addad and Li in [19]. They considered the capacitated clustering problem, where for each center c a capacity value U_c is given, and if the center c is chosen, at most U_c points can be assigned to c. They analyzed Chen's algorithm and showed that for any center C, the coreset approximately preserves the optimal capacitated clustering cost. One crucial idea they use in their proof is representation of assignments through network flow. Suppose we are given a fixed set of centers and weighted input points, and we would like to compute a minimum cost assignment of the points to the centers such that the capacities are not violated. This problem can be modeled as a minimum cost network flow problem.

The first hurdle to adapt the approach in [19] is that it is not possible to represent the assignment problem for fair clustering as a simple flow computation problem. Thus it is not clear how to directly use their approach for fair clustering. Nonetheless, we show that for a fixed constraint M, the assignment problem can be modeled in the desired way. Thus, we can get high probability bound w.r.t. a fixed constraint M. However, to obtain a coreset for fair clustering we need to show this w.r.t. all such constraints (and this leads us towards a universal coreset). The number of such constraints can be as large as $n^{\Omega(k\ell)}$. Hence, to obtain the h.p. bound over all M, we need to show that for a fixed M the error probability is at most $1/n^{\Omega(k\ell)}$. However, it is not clear how to show such a strong bound ($1/n^{\Omega(k)}$ bound can be shown). Nevertheless, we show that it is not necessary to consider all those choices of the constraints together – one can focus on a single color and the constraints w.r.t. that color only. Indeed, this is the reason that we apply Chen's algorithm to different color classes independently. Unfortunately, we pay a heavy toll for this: the coreset size is proportional to ℓ, unlike the vanilla coreset size, and it is not clear how to avoid this dependency. Anyway, this solves our problem, as now we have only $n^{\Omega(k)}$ constraints. It follows that, for a fixed color and a fixed constraint matrix, one can apply an approach similar to the one in [19] (the details are slightly different). This allows us to adapt the ideas from [19] and [14] to construct coresets for much more general clustering problems. Next, we describe the details. We will prove the following lemma.

▶ **Lemma 5.** *For any fixed set C of k centers and for all $k \times \ell$ matrices M, w.p. at least $1 - 1/n^{k+2}$, $|\text{cost}(P, M, C) - \text{wcost}(W, M, C)| \le \sum_{(i,j)} \epsilon |P_{i,j}| \cdot 2^j \mu$.*

Now, consider all the rings $B'_{i,j}$ with $j = 0$. Then,

$$\sum_{(i,j):j=0} \epsilon |P_{i,j}| \cdot 2^j \mu \le \epsilon n \cdot \mu \le \epsilon \cdot \text{OPT}_v \le \epsilon \cdot \text{cost}(P, M, C).$$

Here, OPT_v is the optimal cost of vanilla k-median clustering. The last inequality follows, as the optimal cost of vanilla clustering is at most the cost of any constrained clustering. On the other hand, for any ring $B'_{i,j}$ with $j \ge 1$ and any point p in the ring, $d(p, c_i^*) \ge 2^{j-1}\mu$. Thus,

$$\sum_{(i,j):j\ge 1} \epsilon |P_{i,j}| \cdot 2^j \mu \le \epsilon \sum_{p \in P} 2 \cdot d(p, c_{i_p}^*) \le 2\epsilon \cdot \text{OPT}_v \le 2\epsilon \cdot \text{cost}(P, M, C),$$

where for a point $p \in P$ by i_p we denote the index of a center such that p belongs to $B'_{i_p,j}$ for some j.

Taking union bound over all C and scaling down ϵ by 3 factor, we get the desired result.

▶ **Lemma 6.** *For every set C of k centers and every $k \times \ell$ matrices M, w.p. at least $1 - 1/n$, $|\text{cost}(P, M, C) - \text{wcost}(W, M, C)| \le \epsilon \cdot \text{cost}(P, M, C)$.*

3.2 Proof of Lemma 5

Let P_τ be the points in P of color τ. Also, let W_τ be the chosen samples of color τ. For $1 \leq t \leq \ell - 1$, let $W^t = (\cup_{\tau=1}^{t} W_\tau) \cup (\cup_{\tau=t+1}^{\ell} P_\tau)$. Also, let $W^\ell = \cup_{\tau=1}^{\ell} W_\tau$ be the coreset points of all colors. Recall that for any ring $B'_{i,j}$, $P'_{i,j,\tau}$ is the points of color τ in the ring. Also, $P_{i,j} = \cup_{\tau=1}^{\ell} P'_{i,j,\tau}$.

Note that in the above, W^t contains the sampled points for color 1 to t and original points of color $t+1$ to ℓ. We will prove the following lemma that gives a bound when the coreset contains sampled points of a fixed color t and original points of the other colors.

▶ **Lemma 7.** *Consider any color $1 \leq t \leq \ell$. For any fixed set C of k centers and for all $k \times \ell$ matrices M, w.p. at least $1 - 1/n^{k+4}$, $|\text{cost}(P, M, C) - \text{wcost}(W_t \cup (P \setminus P_t), M, C)| \leq \sum_{(i,j)} \epsilon |P'_{i,j,t}| \cdot 2^j \mu$.*

Note that for a particular color class, if we select all original points in the coreset, there is no error corresponding to those coreset points. This is true, as one can use the corresponding optimal assignment for these points. Assuming that the above lemma holds, now we prove Lemma 5. Consider the coreset W^1. From the above lemma, we readily obtain the following.

▶ **Corollary 8.** *For any fixed set C of k centers and for all $k \times \ell$ matrices M, w.p. at least $1 - 1/n^{k+4}$, $|\text{cost}(P, M, C) - \text{wcost}(W^1, M, C)| \leq \sum_{(i,j)} \epsilon |P'_{i,j,1}| \cdot 2^j \mu$.*

Now, in W^1 consider replacing the points of P_2 by the samples in W_2. We obtain the coreset W^2. Note that the samples in W_1 and W_2 are chosen independent of each other. Thus, by taking union bound over colors 1 and 2, from Lemma 7 we obtain, for all M, w.p. $\geq 1 - 2/n^{k+4}$, $|\text{cost}(P, M, C) - \text{wcost}(W^2, M, C)| \leq \sum_{(i,j)} \epsilon(|P'_{i,j,1}| + |P'_{i,j,2}|) \cdot 2^j \mu$. Similarly, by taking union bound over all $\ell \leq n$ colors and noting that $W^\ell = W$, Lemma 5 follows.

3.3 Proof of Lemma 7

Recall that P_t is the set of points of color t, and W_t is the coreset points of color t. C is the given set of centers. For any matrix M, let M^t be the t^{th} column of M. We have the following observation that implies that it is sufficient to consider the points only in P_t to give the error bound.

▶ **Observation 9.** *Suppose w.p. at least $1 - 1/n^{k+4}$, for all column matrices M', $|\text{cost}(P_t, M', C) - \text{wcost}(W_t, M', C)| \leq \sum_{(i,j)} \epsilon |P'_{i,j,t}| \cdot 2^j \mu$. Then, with the same probability, for all $k \times \ell$ matrices M, $|\text{cost}(P, M, C) - \text{wcost}(W_t \cup (P \setminus P_t), M, C)| \leq \sum_{(i,j)} \epsilon |P'_{i,j,t}| \cdot 2^j \mu$.*

Proof. Consider any $k \times \ell$ matrix M. Then,

$$\text{cost}(P, M, C) = \sum_{\tau=1}^{\ell} \text{cost}(P_\tau, M^\tau, C), \text{ and}$$

$$\text{wcost}(W_t \cup (P \setminus P_t), M, C) = \text{wcost}(W_t, M^t, C) + \sum_{\tau \in [\ell] \setminus \{t\}} \text{cost}(P_\tau, M^\tau, C)$$

It follows that,

$$|\text{cost}(P, M, C) - \text{wcost}(W_t \cup (P \setminus P_t), M, C)| = |\text{cost}(P_t, M^t, C) - \text{wcost}(W_t, M^t, C)|$$

Now, by our assumption, it follows that the probability of the event: for all M, $|\text{cost}(P_t, M^t, C) - \text{wcost}(W_t, M^t, C)|$ exceeds $\sum_{(i,j)} \epsilon |P'_{i,j,t}| \cdot 2^j \mu$ is at most $1/n^{k+4}$. Hence, the observation follows. ◀

By the above observation, it is sufficient to prove that w.p. at least $1 - 1/n^{k+4}$, for all column matrices M, $|\text{cost}(P_t, M, C) - \text{wcost}(W_t, M, C)| \leq \sum_{(i,j)} \epsilon |P'_{i,j,t}| \cdot 2^j \mu$. The proof of this claim is similar to the analysis in [19] and appears in Section 4 of the full version.

The Overlapping Group Case

We are given ℓ groups of points $P_1, \ldots, P_\ell$ such that a point can potentially belong to multiple groups. Note that the algorithm in the disjoint case does not work here. This is because we sample points from each group separately and independently, and thus it is not clear how to assign the weight of a point that belongs to multiple groups. One might think of the following trivial modification of this algorithm. Assign each point to a single group to which it belongs. Based on this assignment, now we have disjoint groups, and we can apply our previous algorithm. But, the new algorithm can have a very large error bound. For example, suppose a point p belongs to two groups i and j, and it is assigned to group i. Also, suppose p was not chosen in the sampling process. Note that the weight of p is represented by some other chosen point p', which was also assigned to group i. However, now we have lost the information that this weight of p was also contributing towards fairness of group j. Thus, the constructed coreset might not preserve any optimal fair clustering with a small error. In the overlapping case, it is not clear how to obtain a coreset whose size depends linearly in ℓ – we design a new coreset construction algorithm where the size depends linearly on Γ. Recall that Γ is the maximum number of distinct collections of groups to which a point belongs.

The main idea of our algorithm is to divide the points into equivalence classes based on their group membership and sample points from each equivalence class. Let $P = \cup_{i=1}^{\ell} P_i$. For each point $p \in P$, let $J_p \subseteq [\ell]$ be the set of indexes of the groups to which p belongs. Let I be the distinct collection of these sets $\{J_p \mid p \in P\}$ and $|I| = \Gamma$. In particular, let $I_1, \ldots, I_\Gamma$ be the distinct sets in I. Now, we partition the points in P based on these sets. For $1 \leq i \leq \Gamma$, let $P^i = \{p \in P \mid I_i = J_p\}$. Thus, $\{P^i \mid 1 \leq i \leq \Gamma\}$ defines equivalence classes for P such that two points $p, p' \in P$ belong to the same equivalence class if they are in exactly the same set of groups. Now we apply our algorithm in the disjoint case on the disjoint sets of points $P^1, \ldots, P^\Gamma$. Let W be the constructed coreset.

Note that here we have Γ disjoint classes, and thus the coreset size is $O(\Gamma(k \log n)^2/\epsilon^3)$. As our coreset size is at least Γ, we assume that $\Gamma < n$. Note that the equivalence classes can be computed in $O(n\ell)$ time, and thus the algorithm runs in time $O(n\ell)+O(nk) = O(n(k+\ell))$. The proof that W is indeed a universal coreset w.h.p. follows the same line as of the disjoint-group case. Again, the idea here is to reduce the analysis to the one class case. However, this is not as straightforward as in the disjoint case. Note that although the classes $P^1, \ldots, P^\Gamma$ are disjoint, two classes can contain points from the same group. Moreover, the constraints are defined w.r.t. the groups, not w.r.t. the classes. Thus, two classes need to interact to satisfy the constraints. Nevertheless, we use the independence of the samples from different classes and exploit the structure of a special class of matrices to complete the proof, which appears in Section 5 of the full version.

The algorithm in the Euclidean case is the same as for general metrics, except we set s to $\Theta(k \log(nb)/\epsilon^3)$, where $b = \Theta(k \log(n/\epsilon)/\epsilon^d)$. Here the main challenge is that it is not possible to take union bound over all possible sets of k centers. Nevertheless, we show that for every set $C \subseteq \mathbb{R}^d$ of k centers and constraint M, the optimal cost is preserved approximately w.h.p. Our main contribution in this part is to devise a discretization technique for obtaining a finite set of centers, so that if instead we draw centers from this set, the cost of any clustering is preserved approximately. The details appear in Section 6 of the full version.

3.4 Approximation Algorithms Based on Universal Coresets

All the algorithms that we design follow one general strategy: first, compute a universal coreset, then, enumerate a small family of sets of possible k centers, such that at least one of them is guaranteed to provide a good approximation, and finally pick the best set of centers by finding the optimal assignment from the coreset to each of the center sets. Throughout this section, we limit our discussion to fair clustering. One major difficulty in the case of fair clustering is solving the assignment problem, for which we devise a novel FPT algorithm.

Solving the Assignment Problem

The fair assignment problem is the following: given an instance of (α, β)-Fair Clustering and a set of k centers C, compute a minimum-cost fair assignment to the centers of C. For fair clustering, the assignment problem is one of the features that makes it harder than other constrained clustering problems. While often the optimal assignment can be found with the help of network flow, e.g. for capacitated clustering, there was no previously known algorithms to compute an optimal or approximate fair assignment without violating the constraints. Moreover, it was observed by Bera et al. [7] that the fair assignment problem is NP-hard, so there is no hope to have a polynomial time assignment algorithm.

We design a fair assignment algorithm with running time $(k\Gamma)^{O(k\Gamma)}n^{O(1)}$. The general idea is to reduce to a linear programming instance. The unknown optimal assignment can be naturally expressed in terms of linear inequalities by introducing a variable f_{ij} for the i-th point and the j-th center, denoting which fraction of the point is assigned to each center, and constraints $f_{ij} \geq 0$ for all i, j, and $\sum_{j=1}^{k} f_{ij} = 1$. Clearly this generalizes a discrete assignment, which corresponds to exactly one of $\{f_{ij}\}_{j=1}^{k}$ being equal to 1, for each $i \in [n]$. Moreover, the fairness of the assignment can also be expressed as linear constraints on $\{f_{ij}\}$.

The main obstacle is that in general the optimal solution to this linear program is not integral, and the integrality gap could be arbitrarily large. Thus, an optimal fractional solution does not yield the desired assignment, and this is not surprising since the fair assignment problem is NP-hard. One possible solution could be restricting the variables to be integral, solving an integer linear program (ILP) instead. But the number of variables is too large for an FPT algorithm. Instead, we introduce the integral variables $\{g_{tj}\}$ denoting how many points from the t-th point equivalence class gets to the j-th center, while leaving the $\{f_{ij}\}$ variables to be fractional. Thus, we obtain an instance of mixed-integer linear programming (MILP) with $k\Gamma$ integer variables and nk fractional variables. By using the celebrated result of Lenstra [36] with subsequent improvements by Kannan [31], and Frank and Tardos [24], we obtain an optimal solution to the MILP instance in time $(k\Gamma)^{O(k\Gamma)}n^{O(1)}$.

Now we explain that after constraining the $\{g_{tj}\}$ variables to be integral, we can assume that all the other variables $\{f_{ij}\}$ are integral too, thus we actually obtain an optimal discrete assignment of the same cost. Consider a particular point equivalence class P^t, and the integral values $\{g_{tj}\}_{j=1}^{k}$ from the optimal solution to the MILP. When these values are fixed, the problem boils down to finding an assignment from P^t to C such that exactly g_{tj} points are assigned to the j-th center. This problem can be solved by a minimum-cost maximum flow in the network where each point has supply one, the j-th center has demand of g_{tj}, and the costs are the distances between the respective points. Moreover, the values $\{f_{ij}\}$ from the MILP correspond exactly to the flow values on the respective edges. Since there is an optimal integral flow in this network, this flow is also an optimal integral solution for $\{f_{ij}\}$.

The downside of the above algorithm is that the time complexity is roughly n^5, and we cannot use it directly to obtain a near-linear time algorithm. So, we also show how to obtain a $(1+\epsilon)$-approximate fair assignment in near-linear time with the help of the coreset. For this, we compute a universal coreset, and then compute the optimal fair assignment from the coreset to the centers C. However, this does not yet give us a fair assignment of the original points to the centers. To construct this assignment, we take the values $\{g_{tj}\}$ computed by the assignment algorithm on the coreset, and then, for each point equivalence class P^t, we solve the simple assignment problem from P^t to C that assigns exactly g_{tj} points to the j-th center. As mentioned above, this can be done by a network flow algorithm. Since the network is bipartite and one of the parts is small (k), this problem can be solved in near-linear time by the specialized flow algorithm in [3]. By the property of the universal coreset, the resulting assignment achieves a $(1 + \epsilon)$-approximation (see Section 8 of the full version).

$(1 + \epsilon)$-Approximation in $\mathbb{R}^d$. Besides our coreset construction and assignment algorithm, the key ingredient to obtain a $(1 + \epsilon)$-approximation is the generic clustering algorithm of Bhattacharya et al. [10]. Their algorithm outputs a list of $2^{\tilde{O}(k/\epsilon^{O(1)})}$ candidate sets of k centers, such that for any clustering of the points there exists a set of centers C in this list that is slightly worse than the optimal set of centers for this clustering. Together with our assignment algorithm this provides a $(1 + \epsilon)$-approximation algorithm with the running time of $2^{\tilde{O}(k/\epsilon^{O(1)})}(k\Gamma)^{O(k\Gamma)}nd \log n$. We describe this algorithm in Section 9 of the full version.

$(3 + \epsilon)$-Approximation in General Metric. With the help of our universal coreset, the strategy to obtain $(3 + \epsilon)$-approximation for (α, β)-Fair k-median is essentially same as that used in [18, 19]: from each of the clusters in an optimal solution on the coreset we guess the closest point to the center, called a *leader* of that cluster. We also guess a suitably discretized distance from each leader to the center of the corresponding cluster. Finally, selecting any center that has roughly the guessed distance to the leader provides us with a $(3 + \epsilon)$-approximation. In this way we obtain a list of $|W|^k(\log n/\epsilon)^{O(k)}$ candidate sets of k centers. Afterwards, our algorithm proceeds similarly to the Euclidean case above, resulting in the running time of $(k\Gamma)^{O(k\Gamma)}/\epsilon^{O(k)} \cdot n \log n$ (see Section 10 of the full version).

References

1 Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. Clustering without over-representation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 267–275, 2019.

2 Sara Ahmadian and Chaitanya Swamy. Improved approximation guarantees for lower-bounded facility location. In *International Workshop on Approximation and Online Algorithms*, pages 257–271. Springer, 2012.

3 Ravindra K. Ahuja, James B. Orlin, Clifford Stein, and Robert E. Tarjan. Improved algorithms for bipartite network flow, 1994.

4 Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. Scalable fair clustering. In *International Conference on Machine Learning*, pages 405–413, 2019.

5 Daniel Baker, Vladimir Braverman, Lingxiao Huang, Shaofeng H-C Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for clustering in graphs of bounded treewidth. *arXiv preprint*, 2019. arXiv:1907.04733.

6 Luca Becchetti, Marc Bury, Vincent Cohen-Addad, Fabrizio Grandoni, and Chris Schwiegelshohn. Oblivious dimension reduction for k-means: beyond subspaces and the johnson-lindenstrauss lemma. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1039–1050, 2019.

7 Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. In *Advances in Neural Information Processing Systems*, pages 4954–4965, 2019.

8 Suman K. Bera, Deeparnab Chakrabarty, and Maryam Negahbani. Fair algorithms for clustering. *CoRR*, abs/1901.02393, 2019. arXiv:1901.02393.

9 Ioana O Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R Schmidt, and Melanie Schmidt. On the cost of essentially fair clusterings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

10 Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Faster Algorithms for the Constrained k-means Problem. *Theory of Computing Systems*, 62(1):93–115, 2018. doi:10.1007/s00224-017-9820-7.

11 Matteo Böhm, Adriano Fazzone, Stefano Leonardi, and Chris Schwiegelshohn. Fair clustering with multiple colors. *arXiv preprint*, 2020. arXiv:2002.07892.

12 Vladimir Braverman, Dan Feldman, and Harry Lang. New frameworks for offline and streaming coreset constructions. *arXiv preprint*, 2016. arXiv:1612.00889.

13 Vladimir Braverman, Shaofeng H-C Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for clustering in excluded-minor graphs and beyond. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2679–2696. SIAM, 2021.

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

15 Xingyu Chen, Brandon Fain, Liang Lyu, and Kamesh Munagala. Proportionally fair clustering. In *International Conference on Machine Learning*, pages 1032–1041, 2019.

16 Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, pages 5029–5037, 2017.

17 Vincent Cohen-Addad. Approximation schemes for capacitated clustering in doubling metrics. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2241–2259. SIAM, 2020.

18 Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT approximations for k-median and k-means. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 42:1–42:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

19 Vincent Cohen-Addad and Jason Li. On the fixed-parameter tractability of capacitated clustering. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 41:1–41:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

20 Hu Ding and Jinhui Xu. A unified framework for clustering constrained data without locality property. *Algorithmica*, 82(4):808–852, 2020.

21 Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 569–578, 2011.

22 Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering. *SIAM Journal on Computing*, 49(3):601–657, 2020.

23 Gereon Frahling and Christian Sohler. Coresets in dynamic geometric data streams. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 209–217, 2005.

24 András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987. doi:10.1007/bf02579200.

25 Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. *Discrete & Computational Geometry*, 37(1):3–19, 2007.

26 Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. *Discret. Comput. Geom.*, 37(1):3–19, 2007.

27 Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 291–300. ACM, 2004.

28 Lingxiao Huang, Shaofeng Jiang, and Nisheeth Vishnoi. Coresets for clustering with fairness constraints. In *Advances in Neural Information Processing Systems*, pages 7589–7600, 2019.

29 Lingxiao Huang and Nisheeth K Vishnoi. Coresets for clustering in euclidean spaces: Importance sampling is nearly optimal. *arXiv preprint*, 2020. arXiv:2004.06263.

30 Piotr Indyk. Sublinear time algorithms for metric space problems. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 428–434, 1999.

31 Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987. doi:10.1287/moor.12.3.415.

32 Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In *36th International Conference on Machine Learning, ICML 2019*, pages 5984–6003. International Machine Learning Society (IMLS), 2019.

33 Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. Guarantees for spectral clustering with fairness constraints. *arXiv preprint*, 2019. arXiv:1901.08668.

34 Amit Kumar, Yogish Sabharwal, and Sandeep Sen. Linear-time approximation schemes for clustering problems in any dimensions. *J. ACM*, 57(2):5:1–5:32, 2010.

35 Michael Langberg and Leonard J Schulman. Universal ε-approximators for integrals. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 598–607. SIAM, 2010.

36 H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983. doi:10.1287/moor.8.4.538.

37 Clemens Rösner and Melanie Schmidt. Privacy preserving clustering with constraints. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

38 Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. Fair coresets and streaming algorithms for fair k-means. In *International Workshop on Approximation and Online Algorithms*, pages 232–251. Springer, 2019.

39 Christian Sohler and David P Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 802–813. IEEE, 2018.

40 Zoya Svitkina. Lower-bounded facility location. *ACM Transactions on Algorithms (TALG)*, 6(4):1–16, 2010.