

Improved Approximation and Scalability for Fair Max-Min Diversification *

Raghavendra Addanki Andrew McGregor Alexandra Meliou Zafeiria Moumoulidou

Abstract

Given an n -point metric space (\mathcal{X}, d) where each point belongs to one of $m = O(1)$ different categories or groups and a set of integers k_1, \dots, k_m , the fair Max-Min diversification problem is to select k_i points belonging to category $i \in [m]$, such that the minimum pairwise distance between selected points is maximized. The problem was introduced by Moumoulidou et al. [ICDT 2021] and is motivated by the need to down-sample large data sets in various applications so that the derived sample achieves a balance over *diversity*, i.e., the minimum distance between a pair of selected points, and *fairness*, i.e., ensuring enough points of each category are included. We prove the following results:

1. We first consider general metric spaces. We present a randomized polynomial time algorithm that returns a factor 2-approximation to the diversity but only satisfies the fairness constraints in expectation. Building upon this result, we present a 6-approximation that is guaranteed to satisfy the fairness constraints up to a factor $1 - \epsilon$ for any constant ϵ . We also present a linear time algorithm returning an $m + 1$ approximation with exact fairness. The best previous result was a $3m - 1$ approximation.
2. We then focus on Euclidean metrics. We first show that the problem can be solved *exactly* in one dimension. For constant dimensions, categories and any constant $\epsilon > 0$, we present a $1 + \epsilon$ approximation algorithm that runs in $O(nk) + 2^{O(k)}$ time where $k = k_1 + \dots + k_m$. We can improve the running time to $O(nk) + \text{poly}(k)$ at the expense of only picking $(1 - \epsilon)k_i$ points from category $i \in [m]$.

Finally, we present algorithms suitable to processing massive data sets including single-pass data stream algorithms and composable coresets for the distributed processing.

1 Introduction

Given a universe of n elements \mathcal{X} and a metric distance function $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_0^+$, the Max-Min diversification problem seeks to select a k -sized subset \mathcal{S} of \mathcal{X} such that the minimum distance between the points in \mathcal{S} is maximized [22, 47]. Intuitively, the goal is to maximize the *dissimilarity* across all the selected points while k is typically much smaller than n . A considerable amount of work in the database community has addressed the diversity maximization problem in the context of query result diversification [27, 31, 51], efficient indexing schemes for result diversification [5, 28, 53], nearest neighbor search [1], ranking schemes [7, 46], and recommendation systems [2, 14].

Recently, Moumoulidou et al. [45] introduced the *fair* variant of the Max-Min diversification problem. Specifically, the assumption is that the universe of elements \mathcal{X} is partitioned into $m = O(1)$ disjoint categories or groups. Then, the aim is to construct a diverse set of points where each group is sufficiently represented. To this end, the input of the problem includes non-negative

*University of Massachusetts Amherst, {raddanki, mcgregor, ameli, zmoumoulidou}@cs.umass.edu

integers k_1, \dots, k_m and the goal now is to select a subset \mathcal{S} using k_i representatives from each group such that the minimum distance across all points is maximized. **As a concrete example, consider a query over a maps service for finding restaurants around Manhattan at NYC. Then the goal is to present the user with a diversified set of restaurant locations while representing different cuisines in the sample.**

In this work, we improve currently known approximation results for fair Max-Min diversification. This includes improving the approximation factor in the most general case of the problem; significantly decreasing the approximation factor if we slightly relax the fairness constraints; and reducing the approximation factors to arbitrarily close to 1 when the underlying metric is Euclidean. Before presenting our results, we review related work.

1.1 Related Work

The problem of unconstrained diversity maximization, i.e., when the number of groups $m = 1$, is well-studied in the context of facility location, information retrieval, web search and recommendation systems [7, 15, 22, 31, 34, 36, 40, 44, 46, 47, 51]. We refer the interested readers to the following surveys related to the diversification literature [29, 30].

Among popular diversification models are the *distance-based* models. In these models, the diversity of a set of points is modeled via some function defined over pairwise distances. Max-Sum (also known as *remote-clique*) and Max-Min (also known as *remote-edge* or *p-dispersion*) are two of the most well-established distance-based diversification models [41]. In Max-Sum, diversity is defined as the sum of the pairwise distances of points selected in a set, while in Max-Min the diversity of a set is equal to the minimum pairwise distance. For both problems, there are known 2-approximation algorithms, which yield the best approximation guarantee that can be achieved for both problems [11, 14, 47]. There are also recent works on *distance-based* diversity maximization models in the streaming, distributed, and sliding-window models [6, 12, 18, 41].

Contrary to unconstrained diversity maximization, the problem of fair diversity maximization is less studied. To the best of our knowledge, there is a known 2-approximation local search algorithm for fair Max-Sum diversification [2, 13, 14] where fairness is modeled via partition matroids [48]. Recent work also extends the local search approach to distances of negative type [21]. Another recently studied objective called Sum-Min [11] is defined as the sum of distances of all points to their closest point in the set. Bhaskara et al. [11] present an 8-approximation algorithm for Sum-Min under partition matroid constraints.

The most relevant result to our work is due to Moumoulidou et al. [45] that introduced the fair variant for the Max-Min diversification problem that we also study. The proposed fairness objectives have been widely studied by prior work [10, 19, 20, 23, 24, 33, 42, 43, 49, 52, 54, 55, 56], and are based on the definition of group fairness and statistical parity [32]. It is worth noting that there are other definitions for fairness, like individual or causal fairness [35], but these are not the focus of our work. Moumoulidou et al. [45] designed a polynomial time algorithm that achieved a $3m - 1$ -approximation for fair Max-Min diversification. There is also a recent line of work for designing (composable) coresets for various distance-based diversification objectives in the fairness setting [16, 17]. Coresets are small subsets of the original data that contain a good approximate solution and are typically used for speed up purposes or designing streaming and distributed algorithms. Prior efforts leave as an open question the construction of coresets for the fair variant of the Max-Min diversification objective.

1.2 Our Results

We present results for both the cases of general metrics and Euclidean metrics.

1. **General Metrics.** In Section 3.1, we present a randomized polynomial time algorithm that returns a factor 2-approximation to the diversity but only satisfies the fairness constraints in expectation, i.e., for each $i \in [m]$, the output is expected to include at least k_i points from \mathcal{X}_i . In Section 3.2, we present a 6-approximation that is guaranteed to include $(1 - \epsilon)k_i$ points in each group $i \in [m]$ assuming each $k_i = \Omega(\epsilon^{-2} \log m)$. Both these results are based on randomized rounding of a linear program. Finally, in Section 3.3 we present a linear time algorithm returning an $m+1$ approximation with perfect fairness. This is an improvement over the previously known $3m - 1$ approximation [45]. We present an example in Appendix A.1 that shows that the analysis presented in Moumoulidou et al. [45] cannot be improved to obtain a better approximation factor. In Section 3.4, we present a hardness of approximation result arguing that we cannot get an approximation factor better than 2, even allowing for multiplicative approximations in fairness constraints.
2. **Euclidean Metrics.** If the points can be embedded in low dimensional space \mathbb{R}^D (e.g., if the points correspond to geographical locations) and the distances correspond to Euclidean distances then we can significantly improve the approximation factors of our algorithms. In Section 4.1, we show that the problem can be solved *exactly* for $D = 1$. For constant dimensions, groups, we then present a $1+\epsilon$ approximation algorithm that runs in $O(nk) + 2^{O(k)}$ time where $k = k_1 + k_2 + \dots + k_m$. In Section 4.3, we show how to improve the running time to $O(nk) + \text{poly}(k)$ at the expense of only picking $(1 - \epsilon)k_i$ points from group $i \in [m]$. All these results are based on a new coresets construction.

In Sections 5.1 and 5.2, we present algorithms suitable to processing massive data sets including single-pass data stream algorithms and composable coresets for distributed processing.

2 Background and Preliminaries

2.1 Fair Max-Min Diversification

We formally define the problem of fair Max-Min diversification recently introduced in [45].

Definition 1 (FAIR MAX-MIN). *Let (\mathcal{X}, d) be a metric space where $\mathcal{X} = \bigcup_{i=1}^m \mathcal{X}_i$ is a universe of n elements partitioned into m non-overlapping groups and $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_0^+$ is a metric distance function. Then $\forall u, v \in \mathcal{X}$, d satisfies the following properties: (1) $d(u, v) = 0$ iff $u = v$ (identity), (2) $d(u, v) = d(v, u)$ (symmetry), and (3) $d(u, v) \leq d(u, w) + d(w, v)$ (triangle inequality). Further, let k_1, k_2, \dots, k_m be non-negative integers with $k_i \leq |\mathcal{X}_i|$, $\forall i \in [m]$. The problem of fair Max-Min diversification is now defined as follows:*

$$\begin{aligned} & \underset{\mathcal{S} \subseteq \mathcal{X}}{\text{maximize}} && \min_{\substack{u, v \in \mathcal{S} \\ u \neq v}} d(u, v) \\ & \text{subject to} && |\mathcal{S} \cap \mathcal{X}_i| = k_i, \forall i \in [m] \quad (\text{fairness constraints}) \end{aligned}$$

The aim is to select a subset $\mathcal{S} \subseteq \mathcal{X}$ of points that maximizes the minimum pairwise distance across the points in \mathcal{S} while being constrained to include k_i points from group i . Throughout the paper we refer to the diversity of a set \mathcal{S} as $\text{div}(\mathcal{S}) = \min_{u, v \in \mathcal{S}, u \neq v} d(u, v)$.

Let $\mathcal{S}^* = \bigcup_{i=1}^m \mathcal{S}_i^*$ be the set of points that obtains the optimal diversity score denoted by $\text{div}(\mathcal{S}^*) = \ell^*$. We say a subset of points \mathcal{S} is an α approximation if $\text{div}(\mathcal{S}) \geq \ell^*/\alpha$ and achieves β fairness if $|\mathcal{S} \cap \mathcal{X}_i| \geq \beta k_i$ for all $i \in [m]$. When $\beta = 1$, we say subset achieves *perfect fairness*.

FAIR MAX-MIN is an NP-hard problem for which the best known polynomial time algorithms are: a 4-approximation algorithm that only works for $m = 2$ groups and a $3m - 1$ -approximation algorithm that yields the best guarantees for any $m \geq 3$ [45]. The best approximation factor one can hope for in general metric spaces is a 2-approximation guarantee. This claim easily follows since when $m = 1$, the problem is just the Max-Min diversification problem where it is known that no polynomial time algorithm with an approximation factor better than 2 exists if $P \neq NP$ [47]. We use $\text{poly}(\cdot)$ to describe polynomial time algorithms using the context dependent parameters.

2.2 Low Doubling Dimension Spaces

Our results for low dimensional Euclidean metrics use the fact that such metrics have *low doubling dimension*. Our work in this direction is inspired by work on diversity maximization by Ceccarello et al. [16, 17, 18]. We define a ball of radius r centered at $p \in \mathcal{X}$ as the set of all points in \mathcal{X} within distance strictly less than r from p . We use the notation: $\mathbf{B}(p, r) = \{q \in \mathcal{X} \mid d(p, q) < r\}$.

Definition 2 (DOUBLING DIMENSION). *Let (\mathcal{X}, d) be a metric space. The doubling dimension of \mathcal{X} is the smallest integer λ such that any ball $\mathbf{B}(p, r)$ of radius r around a point $p \in \mathcal{X}$ can be covered using at most $(r/r')^\lambda$ balls of radius r' . The Euclidean metric on \mathbb{R}^D has doubling dimension $O(D)$ [9, 18, 39].*

2.3 Coresets

Coresets are powerful theoretical tools for designing efficient optimization algorithms in the presence of massive datasets in sequential, streaming or distributed environments [3, 41]. At a high level, coresets are carefully chosen subsets of the original universe of elements that contain an approximate solution to the optimal solution for the optimization problem at hand. A coreset for fair Max-Min diversification is defined as follows:

Definition 3 (CORESET FOR FAIR MAX-MIN). *A set $\mathcal{T} \subseteq \mathcal{X}$ is an α -coreset if there exists a subset $\mathcal{T}' \subseteq \mathcal{T}$ with $|\mathcal{T}' \cap \mathcal{X}_i| = k_i \forall i \in [m]$ and $\text{div}(\mathcal{T}') \geq \ell^*/\alpha$.*

Note that optimally solving FAIR MAX-MIN on \mathcal{T} , a set typically much smaller in size than \mathcal{X} , yields an α -approximation factor. Further, the notion of coresets is useful for designing algorithms in the distributed setting using the *composability* property. *Composable coresets* closely relate to the notion of *mergeable summaries* [4, 41] while the assumption is that the universe of elements \mathcal{X} is partitioned into L subsets (e.g., processing sites). Then the goal is to process each subset independently and extract a *local* coreset such that in the union of these local coresets, there is an approximate solution for the optimization problem at hand. Specifically, for FAIR MAX-MIN a composable coreset is defined as follows:

Definition 4 (COMPOSABLE CORESET FOR FAIR MAX-MIN). *A function $c(\mathcal{X})$ that maps a set of elements to a subset of these elements computes an α -composable coreset for some $\alpha \geq 1$, if for any partitioning¹ of $\mathcal{X} = \bigcup_j \mathcal{Y}_j$ and $\mathcal{T} = \bigcup_j c(\mathcal{Y}_j)$, there exists a set $\mathcal{T}' \subseteq \mathcal{T}$ with $|\mathcal{T}' \cap \mathcal{X}_i| = k_i \forall i \in [m]$ such that $\text{div}(\mathcal{T}') \geq \ell^*/\alpha$.*

¹The notion of composable coresets can also be extended when \mathcal{X} is not divided into disjoint subsets but this is not the focus of our work.

3 General Metrics

In this section, we present algorithms for FAIR MAX-MIN with an arbitrary metric. Our first two algorithms are based on rounding a suitable linear program. In Section 3.3 we present a linear time algorithm returning an $m + 1$ approximation with perfect fairness. Finally, in Section 3.4, we give hardness of approximation results for FAIR MAX-MIN.

3.1 2-Approx with Expected Fairness

In this section and others, we assume a guess γ on the optimal diversity value for FAIR MAX-MIN. Note there are at most $\binom{n}{2}$ possible values for the optimal diversity corresponding to the set of distances between pairs of points. Hence, trying all these guesses only increases the running time by a factor $O(n^2)$. Assuming the ratio between the largest and smallest distance is $\text{poly}(n)$, this can be reduced to $O(\epsilon^{-1} \log n)$ at the expense of introducing an additional factor of $1 + \epsilon$ in the approximation. This follows by the standard technique of only considering guesses that are powers of $(1 + \epsilon)$ [38].

Fair Max-Min LP. Let $\mathcal{X} = \{p_1, \dots, p_n\}$. For every point $p_j \in \mathcal{X}$, we have a variable x_j . We represent the fairness constraint for every group $i \in [m]$ using constraint (1). Additionally, for every point $p \in \mathcal{X}$, we add the constraint (2) that includes at most one point in a ball of radius $\gamma/2$ centered at p . This ensures that the selected points are separated by a distance of at least $\gamma/2$. Using constraint (3), we allow x_p to take any value between 0 and 1. If $\gamma \leq \ell^*$, observe that the optimal solution for FAIR MAX-MIN is a feasible solution for this LP.

$$\sum_{p_j \in \mathcal{X}_i} x_j \geq k_i \quad \forall i \in [m]. \quad (1)$$

$$\sum_{p_\ell \in \mathbf{B}(p, \gamma/2)} x_\ell \leq 1 \quad \forall p \in \mathcal{X}. \quad (2)$$

$$x_j \geq 0 \quad \forall j \in [n]. \quad (3)$$

Let x_j^* denote the optimal solution of the linear program stated above. Let $n' = |\{j : x_j^* > 0\}|$ and without loss of generality suppose $x_j^* > 0$ for all $j \in [n']$. We obtain an integral solution using a randomized rounding algorithm, in which we generate a random ordering based on sampling without replacement, such that a point p_j is selected as the next point in the ordering with probability proportional to x_j^* . This allow us to show (see Lemma 1) that the rounding scheme returns a set \mathcal{S} with at least k_i points *in expectation* from each group $i \in [m]$ (satisfying constraint (1) in expectation). Further, our rounding scheme selects at most one point from each ball of radius $\gamma/2$ (satisfying constraint (2)). Since for a $\gamma \leq \ell^*$ there is a set \mathcal{S} that satisfies the properties discussed above, selecting the set \mathcal{S} for the largest guess γ results in a 2-approximation for the diversity score.

Randomized Rounding. We generate a random ordering σ of $[n']$ where $\sigma(t)$ is randomly chosen from $R_t = [n'] \setminus \{\sigma(1), \dots, \sigma(t-1)\}$ such that for $j \in R_t$, $\Pr[\sigma(t) = j] = \frac{x_j^*}{\sum_{\ell \in R_t} x_\ell^*}$. After generating the ordering σ , we construct the output set \mathcal{S} by including the point p_j in \mathcal{S} iff $\sigma(j) \leq \sigma(\ell)$ for all $p_\ell \in \mathbf{B}(p_j, \gamma/2)$. Note that all points in the output are at least distance $\gamma/2$ apart.

Lemma 1. *There is an algorithm that returns a set \mathcal{S} , such that for all groups $i \in [m]$, it holds that $\mathbb{E}[|\mathcal{S} \cap \mathcal{X}_i|] \geq k_i$. Further all the points selected in \mathcal{S} are at least $\gamma/2$ far apart.*

Proof. Consider the randomized rounding algorithm described in this section. Now, let p_j be a point with $x_j^* > 0$. Define A_t to be the event $d(p_{\sigma(t)}, p_j) < \gamma/2$ and $d(p_{\sigma(t')}, p_j) \geq \gamma/2$ for all $t' < t$.

In other words, A_t is the event that the first point included in \mathcal{S} from the ball $\mathbf{B}(p_j, \gamma/2)$ is the point from the t -th step (in the ordering σ). Then,

$$\begin{aligned} \Pr[p_j \in \mathcal{S}] &= \sum_{t=1}^{n'} \Pr[\sigma(t) = j | A_t] \Pr[A_t] = \sum_{t=1}^{n'} \frac{x_j^*}{\sum_{p_\ell \in \mathbf{B}(p_j, \gamma/2)} x_\ell^*} \Pr[A_t] \\ &= \frac{x_j^*}{\sum_{p_\ell \in \mathbf{B}(p_j, \gamma/2)} x_\ell^*} \sum_{t=1}^{n'} \Pr[A_t] \\ &= \frac{x_j^*}{\sum_{p_\ell \in \mathbf{B}(p_j, \gamma/2)} x_\ell^*} \geq x_j^* \end{aligned}$$

where the last equality follows because $\sum_{t=1}^{n'} \Pr[A_t] = 1$ and the last inequality holds because of constraint (2) in the FAIR MAX-MIN LP. Then for $i \in [m]$, we have $\mathbb{E}[|\mathcal{S} \cap \mathcal{X}_i|] \geq \sum_{p \in \mathcal{X}_i} x_p^* \geq k_i$ where the last inequality follows from constraint (1). \square

3.2 6-Approx with $(1 - \epsilon)$ Fairness

We now present a more involved rounding scheme of the LP given in the previous section that ensures that the selected points contain at least $(1 - \epsilon)k_i$ points in \mathcal{X}_i for each $i \in [m]$. However, this guarantee comes at the expense of increasing the approximation factor for the diversity score from 2 to 6.

The main idea behind the new rounding scheme stems from the observation that for any $p_i, p_j \in \mathcal{X}$, if $\mathbf{B}(p_i, \gamma/2)$ and $\mathbf{B}(p_j, \gamma/2)$ are disjoint, then, in the previous rounding scheme, the event that p_i is included in the returned solution is independent of the event that p_j is included. This follows because the relative ordering of the elements in $\{\ell : p_\ell \in \mathbf{B}(p_i, \gamma/2)\}$ in σ is independent of the ordering of the elements in $\{\ell : p_\ell \in \mathbf{B}(p_j, \gamma/2)\}$ in σ . This independence will ultimately allow us to use Chernoff bound to argue concentration of the number of elements chosen from each group $\mathcal{X}_j \forall j \in [m]$.

3.2.1 Randomized Rounding with improved fairness guarantees

We solve the LP in Section 3.1 to get a feasible solution $\{x_j^*\}_{j \in [n]}$. Next, we transform $\{x_j^*\}_{j \in [n]}$ into a feasible solution $\{y_j^*\}_{j \in [n]}$ for the following set of constraints, some of which are no longer linear:

$$\sum_{p_j \in \mathcal{X}_i} y_j \geq k_i \quad \forall i \in [m]. \quad (1')$$

$$\sum_{p_\ell \in \mathbf{B}(p, \gamma/6)} y_\ell \leq 1 \quad \forall p \in \mathcal{X}. \quad (2')$$

$$y_j \geq 0 \quad \forall j \in [n]. \quad (3')$$

$$(0 < y_i \text{ and } 0 < y_j) \Rightarrow d(p_i, p_j) \geq \frac{\gamma}{3} \quad \forall p_i, p_j \in \mathcal{X}_\ell, \forall \ell \in [m] \quad (4')$$

The constraint (2') ensures that at most one point in a ball of radius $\gamma/6$ is selected (instead of $\gamma/2$ used in Section 3.1) and results in an approximation factor of 6. The constraint (4') ensures that points from the same group with non-zero values are separated by at least $\gamma/3$, which is used to argue $(1 - \epsilon)$ fairness (see Theorem 1). The transformation of x^* to y^* can be done by redistributing the values as follows:

(a) For each $p_j \in \mathcal{X}$ with $x_j^* > 0$ satisfying $p_j \in \mathcal{X}_i$ and y_j^* value not yet set, we set:

$$y_j^* \leftarrow \left(\sum_{p_\ell \in \mathbf{B}(p_j, \gamma/3) \cap \mathcal{X}_i} x_\ell^* \right) \text{ and } y_\ell^* \leftarrow 0 \text{ for all } p_\ell \in \mathbf{B}(p_j, \gamma/3) \cap (\mathcal{X}_i \setminus \{p_j\}).$$

(b) Finally, for all $p_j \in \mathcal{X}$ with $x_j^* = 0$, we set $y_j^* \leftarrow 0$.

Informally, we are just moving weight to p_j from points of the same group (as p_j) that are at a distance strictly less than $\gamma/3$ from p_j .

Lemma 2. $\{y_j^*\}_{j \in [n]}$ satisfies Constraints (1'-4').

Proof. Observe that $\{y_j^*\}_{j \in [n]}$ satisfies the constraint (4'). If a point $p_j \in \mathcal{X}_i$ satisfies $y_j^* > 0$, then, it means that we set y_ℓ^* to 0 for every $p_\ell \in \mathbf{B}(p_j, \gamma/3) \cap (\mathcal{X}_i \setminus \{p_j\})$.

Constraint (2') is satisfied because

$$\sum_{p_\ell \in \mathbf{B}(p_j, \gamma/6)} y_\ell^* \leq \sum_{p_\ell \in \mathbf{B}(p_j, \gamma/6 + \gamma/3)} x_\ell^* = \sum_{p_\ell \in \mathbf{B}(p_j, \gamma/2)} x_\ell^* \leq 1$$

since $\{x_\ell^*\}_{\ell \in [n]}$ satisfies constraint (2). Constraint (1') is satisfied because $\sum_{p_j \in \mathcal{X}_i} y_j^* = \sum_{p_j \in \mathcal{X}_i} x_j^*$ and Constraint (3') is trivially satisfied. \square

We next pick a random permutation σ as in the previous Section 3.1, but now using the values $\{y_\ell^*\}_{\ell \in [n]}$. We add p_j to the output \mathcal{S} if $\sigma(j) \leq \sigma(\ell)$ for all p_ℓ such that $d(p_\ell, p_j) < \gamma/6$. Note that all points in \mathcal{S} are therefore at least a distance of $\gamma/6$ apart.

Theorem 1. Assume $k_i \geq 3\epsilon^{-2} \log(2m)$ for all $i \in [m]$. There is a $\text{poly}(n, k, \delta^{-1})$ time algorithm that returns a subset of points with diversity $\ell^*/6$ and includes $(1-\epsilon)k_i$ points in each group $i \in [m]$ with probability at least $1 - \delta$.

Proof. Let $Y_p = 1$ if the point $p \in \mathcal{X}$ is included in the output \mathcal{S} . Fix $i \in [m]$. The proof of Lemma 1 applied to balls of radius $\gamma/6$ rather than balls of radius $\gamma/2$, ensures that for each $i \in [m]$, $\mathbf{E}[\sum_{p \in \mathcal{X}_i} Y_p] \geq k_i$. The fact $\{Y_p\}_{p \in \mathcal{X}_i}$ are fully independent allows us to apply the Chernoff bound and conclude $\Pr[\sum_{p \in \mathcal{X}_i} Y_p \leq (1-\epsilon)k_i] \leq \exp(-\epsilon^2 k_i/3) \leq 1/(2m)$. Hence, by an application of the union bound, **we ensure that with probability at least 1/2, $|\mathcal{S} \cap \mathcal{X}_i| \geq (1-\epsilon)k_i$ for all $i \in [m]$.** Repeating the process $\log \delta^{-1}$ times ensures that at least one of the trials succeeds with probability at least $1 - \delta$. \square

Note that Theorem 1 requires the k_i values to be sufficiently large, and such conventions have also been used in prior work [11]. For small k_i values, i.e., $k_i = o(\log n)$, the FAIR-GMM algorithm introduced in Moumoulidou et al. [45] obtains a 5-approximation guarantee in polynomial time. Using an additive Chernoff bound, alternatively, we can find at least $k_i - O(\sqrt{k_i \log m})$ points from each group $i \in [m]$, without the requirement of having large k_i 's.

3.3 $(m+1)$ -Approx with Perfect Fairness

We now describe FAIR-GREEDY-FLOW, an $m+1$ -approximation algorithm that ensures perfect fairness. This is an improvement over the previously known $3m-1$ approximation [45]. Specifically, we give the pseudocode for FAIR-GREEDY-FLOW (Algorithm 4) and show that FAIR-GREEDY-FLOW returns a solution which is an $m+1$ -approximation for FAIR MAX-MIN. We also present

an example in Appendix A.1 that shows that the analysis presented in Moumoulidou et al. [45] cannot be improved to obtain a better approximation factor. Missing details from this section are presented in Appendix A.

Overview of Fair-Greedy-Flow. We assume a guess γ for ℓ^* . The algorithm proceeds by iteratively building clusters of close points of distinct groups. Our main idea is to select one point from each cluster such that the fairness constraints are guaranteed. First, we describe the procedure for building a cluster. Let D denote a cluster initialized with a point of group $i \in [m]$. Among the available points \mathcal{R} , we include a point $p \in \mathcal{R}$, if it is within a distance of $\frac{\gamma}{m+1}$ to some point $x \in D$, and no other point of the same group is already present in D .

If there is no such point, the cluster D is complete, and we remove all points from \mathcal{R} that are within a distance of $\frac{\gamma}{m+1}$ from some point in D . Also, we discard all points of group i , i.e., \mathcal{X}_i from \mathcal{R} , as soon as there are at least k distinct clusters in \mathcal{C} containing points from \mathcal{X}_i . We continue this process of iteratively building clusters, until there are points from each group that are part of at least k distinct clusters or if there are no remaining points.

Next, we use an approach similar to [45] and select at most one point from each cluster, satisfying the fairness constraints. We construct a flow network with clusters D_1, D_2, \dots, D_t in \mathcal{C} represented by nodes v_1, v_2, \dots, v_t and groups represented by nodes u_1, u_2, \dots, u_m . We add an edge with capacity 1 between every pair u_i and v_j if there is a point of group i in cluster D_j for some $j \in [t]$. We create a source node a and add edges with capacity k_i between a and $u_i \forall i \in [m]$. We then create a sink node b and add edges with capacity 1 between b and $v_j \forall j \in [t]$. Finally, we find maximum flow using Ford-Fulkerson algorithm [25]. For each edge (u_i, v_j) with flow equal to 1, we include the point of group i from cluster D_j in our solution. We conclude this section with the following theorem:

Theorem 2. FAIR-GREEDY-FLOW Algorithm returns an $(m+1)(1+\epsilon)$ -approximation and achieves perfect fairness for the FAIR MAX-MIN problem using a running time of $O(nkm^3\epsilon^{-1} \log n)$.

3.4 Hardness of Approximation

In this section, we give a hardness of approximation result for the FAIR MAX-MIN problem. Our result is a generalization and improvement over the 2-approximation hardness shown in [45], as we also allow for approximations in fairness constraints.

Definition 5 (GAP-CLIQUE $_\rho$). Given a constant $\rho \geq 1$, a graph G , and an integer k , we want to distinguish between the case where a clique exists of size k (the “yes” case) and the case where no clique exists of size $\geq k/\rho$ (the “no” case).

It is known that GAP-CLIQUE $_\rho$ is NP-hard for every $\rho \geq 1$ [8]. Now, via a reduction from the GAP-CLIQUE $_\rho$ we argue that FAIR MAX-MIN cannot be approximated to a factor better than 2, even allowing for multiplicative approximations in fairness constraints. See Appendix B for the proof of the following theorem.

Theorem 3. Let $\alpha < 2$ and $\beta > 0$ be constants. Unless $P = NP$, there is no polynomial time algorithm for the FAIR MAX-MIN problem that obtains an α -approximation factor for diversity score, and β fairness.

4 Euclidean Metrics

In this section, we assume that the metric space is Euclidean, i.e., we can associate a point $p_i \in \mathbb{R}^D$ with the i th entry of \mathcal{X} and $d(p_i, p_j) = \|p_i - p_j\|_2 = \sqrt{\sum_{\ell \in [D]} (p_i(\ell) - p_j(\ell))^2}$. When $D = 1$

Algorithm 1 FAIR-LINE: An exact algorithm for data on a line

Input: $\mathcal{X} = \bigcup_{i=1}^m \mathcal{X}_i$: Universe of available points.

$k_1, \dots, k_m \in \mathbb{Z}^+$.

$\gamma \in \mathbb{R}^+$: A guess of the optimum fair diversity.

Output: k_i points in \mathcal{X}_i for $i \in [m]$.

```

1: Let  $n \leftarrow |\bigcup_{i=1}^m \mathcal{X}_i|$  and initialize  $H \in \{0, 1\}^{(k_1+1) \times \dots \times (k_m+1) \times n}$  to 0.
2: Set  $H[0, \dots, 0, 0] \leftarrow 1$ ,  $H[0, \dots, 0, 1] \leftarrow 1$ , and if  $p_1 \in \mathcal{X}_\ell$ ,  $H[0, \dots, \underbrace{1}_{\text{index } \ell}, \dots, 0, 1] \leftarrow 1$ .
3: for  $j = 2$  to  $n$  do
4:   Let  $i \in [m]$  satisfy  $p_j \in \mathcal{X}_i$ .
5:   Let  $j' = \max(\{0\} \cup \{j' \in [n] : p_{j'} + \gamma \leq p_j\})$ .
6:   for  $k'_1 \in \{0, \dots, k_1\}, \dots, k'_m \in \{0, \dots, k_m\}$  do
7:      $H[k'_1, \dots, k'_m, j] \leftarrow H[k'_1, \dots, k'_m, j-1]$ .
8:     If  $k'_i \geq 1$ ,  $H[k'_1, \dots, k'_m, j] \leftarrow H[k'_1, \dots, k'_i - 1, \dots, k'_m, j'] \vee H[k'_1, \dots, k'_m, j-1]$ .
9:   end for
10: end for
11: return  $H[k_1, k_2, \dots, k_m, n]$ .
```

we show that the problem can be solved exactly in polynomial time via Dynamic Programming. More generally, when $D = O(1)$ we present a bi-criteria approximation that uses an extension of the dynamic programming approach and properties of low dimensional Euclidean spaces. Missing details from this section are presented in Appendix C.

4.1 Exact Computation in One Dimension

In this section, we assume the points in the universe $\mathcal{X} = \bigcup_{i=1}^m \mathcal{X}_i$ can be embedded on a line. Specifically, let $\mathcal{X} = \{p_1, \dots, p_n\}$ where each $p_i \in \mathbb{R}$ and we order the points such that $p_1 \leq p_2 \leq \dots \leq p_n$. We further assume a guess γ on the optimal diversity score for FAIR MAX-MIN and design the dynamic programming algorithm FAIR-LINE (Algorithm 1) that computes an exact solution when $\gamma = \ell^*$. See the previous section for a discussion on guessing γ .

Dynamic Programming. Define the dynamic programming table $H \in \{0, 1\}^{(k_1+1) \times \dots \times (k_m+1) \times n}$ indexed from 0. An entry $H[k'_1, k'_2, \dots, k'_m, j] \in \{0, 1\}$ is 1 iff there is a subset \mathcal{S}' of the first j points on the line with diversity γ that contains k'_i points from each group $i \in [m]$. To compute the entries of H , we process the points in their order of appearance on the line.

Note that there is a set \mathcal{S}' with k'_i points from each group i among the first j points if: (1) there is such a set among the first $j-1$ points, or (2) point j belongs to group i for some $i \in [m]$, and among the first j' points there is a set with $k'_1, \dots, k'_i - 1, \dots, k'_m$ points from the corresponding groups where $j' < j$ is the largest value such that $d(p_j, p_{j'}) \geq \gamma$.

See FAIR-LINE (Algorithm 1) for the resulting algorithm. For simplicity, the algorithm is written to only determine whether it is possible to pick a subset with diversity γ subject to the required fairness constraints. However, the algorithm can be easily extended to construct a subset of points for every non-zero entry in H by storing a pointer to the choice we made. For an entry $H[k'_1, k'_2, \dots, k'_m, j] = 1$ that also satisfies $H[k'_1, k'_2, \dots, k'_m, j-1] = 1$ we store a pointer to that entry. In the second case, if $H[k'_1, k'_2, \dots, k'_m, j'] = 1$ for some j' , we store a pointer to that entry. We construct the solution set using the stored pointers, starting at $H[k_1, k_2, \dots, k_m, n]$ and backtracking, to indicate which points to add to the solution.

Theorem 4. *There is an algorithm that solves the FAIR MAX-MIN problem exactly when the points can be embedded on a line and requires a running time of $O(n^4 \prod_{i=1}^m (k_i + 1))$.*

4.2 Coresets for Constant Dimensions

In this section, we design efficient $(1+\epsilon)$ -coresets for FAIR MAX-MIN in metric spaces of low doubling dimension (Definition 2). Let λ denote the doubling dimension of \mathcal{X} . Our approach generalizes prior work on constructing efficient coresets for unconstrained Max-Min diversification [18] to the FAIR MAX-MIN problem.

Specifically, we give the first algorithm for constructing coresets in metric spaces of doubling dimension. The proposed approach uses the GMM algorithm that obtains a factor **2-approximation for the unconstrained Max-Min diversification problem** [47, 50].

GMM is a greedy algorithm and works as follows: it starts with an arbitrary point in a set S and in every subsequent step selects the point that is the farthest away from the previously selected points. In fact, readers familiar with the k -center clustering problem will recognize that this is the same strategy used by [37]. If k is the size of the subset to be selected and n is the size of the universe of points, it is known that GMM can be implemented in $O(kn)$ time [43, 53].

Coreset Construction. First, define $\epsilon' = \epsilon/(1+\epsilon)$ and note that $\epsilon/2 \leq \epsilon' < 1$ since $\epsilon \in (0, 1]$. The CORESET Algorithm constructs coreset \mathcal{T} as follows: we run GMM on each group $i \in [m]$ separately to retrieve a set T_i with $O((4/\epsilon')^\lambda k)$ points. The coreset \mathcal{T} is equal to the union of the T_i sets for all $i \in [m]$, namely: $\mathcal{T} \leftarrow \bigcup_{i=1}^m T_i$, where $T_i \leftarrow \text{GMM}(\mathcal{X}_i, (4/\epsilon')^\lambda k)$.

We will show that \mathcal{T} contains a set \mathcal{T}' with $\text{div}(\mathcal{T}') \geq \ell^*/(1+\epsilon)$ and k_i points from each group i . At a high level, the idea is that for each group i there are two cases: (1) either T_i contains a sufficient number of points that are far apart such that even if we had to remove points close to points selected from other groups, we would still have enough points to satisfy fairness, or (2) the optimal points from group i are within small distance from their closest point in T_i . In the analysis we show that in both cases we have enough points from each group i to satisfy fairness while these points are at least $\ell^*/(1+\epsilon)$ far apart.

We first prove the following lemma, which we will use later.

Lemma 3. *Let S be a set of $k' = (4/\epsilon')^\lambda k$ points that are all at least $(\epsilon'/2)\gamma$ far apart. Then, there exists a subset $S' \subset S$ of points that are all at least γ far apart and $|S'| \geq k$.*

Proof. Let $S' = \emptyset$. Add an arbitrary point x from S to S' and remove all points in the ball $\mathbf{B}(x, \gamma)$ from S . Consider a set of balls of radius $(\epsilon'/4)\gamma$ that cover the removed points. Each of these balls cover at most one removed point since discarded points are at least $(\epsilon'/2)\gamma$ far apart. Hence, the number of balls is at least the number of removed points. But because the doubling dimension is λ we know there exists a set of $(4/\epsilon')^\lambda$ balls of radius $(\epsilon'/4)\gamma$ that cover the removed points. Hence, the number of removed points is at most $(4/\epsilon')^\lambda$. Since there were $k' = (4/\epsilon')^\lambda k$ points in S , we may continue in this way until we've added k points to S' . All chosen points are at least γ apart as required. \square

Our main theorem in this section is as follows:

Theorem 5. *There is an algorithm that returns a $(1+\epsilon)$ -coreset of size $O((8/\epsilon)^\lambda km)$ in metrics of doubling dimension λ with a running time $O((8/\epsilon)^\lambda kmn)$.*

Proof. We show that the set $\bigcup_{i=1}^m T_i$ constructed by the CORESET Algorithm is an $(1+\epsilon)$ -coreset by showing the existence of a set $\mathcal{T}' \subseteq \bigcup_{i=1}^m T_i$ with k_i points from each group i and $\text{div}(\mathcal{T}') \geq \ell^*/(1+\epsilon)$.

For every group $i \in [m]$, we define \widehat{T}_i to be the maximal prefix of the points added by GMM to form T_i such $\text{div}(\widehat{T}_i) \geq (\epsilon'/2)\ell^*$. We first process all the groups for which $|\widehat{T}_i| < (4/\epsilon')^\lambda k$, which we call *critical* groups. For all critical groups, any point $p \in \mathcal{X}_i \setminus \widehat{T}_i$ is within distance $(\epsilon'/2)\ell^*$ from its closest point $f(p)$ in \widehat{T}_i , i.e., $d(p, f(p)) < (\epsilon'/2)\ell^*$. As a result, for any pair of optimal points o_1, o_2 in critical groups we deduce:

$$\begin{aligned} d(f(o_1), f(o_2)) &\geq d(o_1, o_2) - d(o_1, f(o_1)) - d(o_2, f(o_2)) \\ &> \ell^* - 2 \cdot \epsilon' \ell^* / 2 = \ell^* / (1 + \epsilon) . \end{aligned}$$

We initialize $\mathcal{T}' = \bigcup_{o \in \bigcup_{i:\text{critical}} \mathcal{S}_i^*} f(o)$ where \mathcal{S}_i^* is the set of points in an optimal solution belonging to group \mathcal{X}_i . We now process all *non-critical* groups $j \in [m]$ in an arbitrary order and remove any point in \widehat{T}_j that is less than ℓ^* apart from some point in \mathcal{T}' . Then we argue that in the remaining points there is a set of points T'_j with k_j points that are at least ℓ^* far apart.

By the doubling dimension property and the fact that all the points in \widehat{T}_j are at least $(\epsilon'/2)\ell^*$ far apart, the removal step described above discards at most $(4/\epsilon')^\lambda \sum_{i:\text{processed groups}} |\mathcal{T}' \cap \mathcal{X}_i|$ points from \widehat{T}_j . Consequently, regardless of the order in which we process the *non-critical* groups, by the time we process \widehat{T}_j for some $j \in [m]$, there will be *at least* $(4/\epsilon')^\lambda k - \sum_{i:\text{processed groups}} (4/\epsilon')^\lambda k_i \geq (4/\epsilon')^\lambda k_j$ points that are at least $(\epsilon'/2)\ell^*$ apart from each other.

Now by applying Lemma 3 on the points of T'_j , we conclude that there are at least k_j points within ℓ^* distance from all other points in \mathcal{T}' . Then this set of points T'_j can be added to \mathcal{T}' to satisfy fairness for group j . Thus, it holds that $\text{div}(\mathcal{T}') \geq \ell^* / (1 + \epsilon)$ which implies the claimed approximation factor for coreset \mathcal{T} .

As $\epsilon' = \epsilon / (1 + \epsilon) \geq \epsilon/2$, we have $|\mathcal{T}| = O((8/\epsilon)^\lambda km)$. Since we use GMM to obtain \mathcal{T} , the running time of the CORESET algorithm is $O((8/\epsilon)^\lambda kmn)$. \square

From the coreset \mathcal{T} , we can obtain a $(1 + \epsilon)$ -approximation by enumerating over all subsets of \mathcal{T} and returning the subset with maximum diversity and perfect fairness. The running time of this algorithm is $O(2^{O(k)} + nk)$, when m, λ are constants. In the next section, we describe an algorithm that has a polynomial dependence on n and k , obtained at the cost of $(1 - \epsilon)$ -fairness.

4.3 $(1 + \epsilon)$ Approx with $(1 - \epsilon)$ Fairness

In this section, we describe FAIR-EUCLIDEAN (Algorithm 3) which uses $(1 + \epsilon)$ -coresets described in Section 4.2 and returns a subset of points with diversity at least $\ell^* / (1 + \epsilon)$ and has $(1 - \epsilon)k_i$ points from each group $i \in [m]$. Missing details are presented in Appendix C.

First, we discuss FAIR-DP (Algorithm 2), which is a dynamic programming subroutine used in FAIR-EUCLIDEAN. The subroutine will be applied to a collection of t disjoint subsets of \mathcal{X} : $\mathcal{C} = \{C_1, C_2, \dots, C_t\}$. This collection will be *well-separated* in the sense that for all $i \neq j$ and $x \in C_i, y \in C_j$ then $d(x, y) \geq \gamma$. Points in the same set can be arbitrarily close together. We design FAIR-DP (Algorithm 2): a dynamic programming algorithm to retrieve a set $\mathcal{F} = \bigcup_{i=1}^m \mathcal{F}_i \subseteq \mathcal{C}$ with k_i points per group i and $\text{div}(\mathcal{F}) \geq \gamma$ if such a set exists in \mathcal{C} .

Dynamic Programming. Define the dynamic programming table $H \in \{0, 1\}^{(k_1+1) \times \dots \times (k_m+1) \times t}$ indexed from 0. An entry $H[k'_1, k'_2, \dots, k'_m, j] \in \{0, 1\}$ is 1 iff there is a subset \mathcal{F}' among the first j clusters such that $|\mathcal{F}' \cap \mathcal{X}_i| \geq k'_i \forall i \in [m]$ and $\text{div}(\mathcal{F}') \geq \gamma$.

To compute the entries of H , we process the clusters in \mathcal{C} using some fixed ordering. Note that there is a set \mathcal{F}' with k'_i points from each group i among the first j clusters if there is a subset

Algorithm 2 FAIR-DP: A dynamic programming subroutine

Input: C_1, C_2, \dots, C_t : Family of disjoint subsets of $\mathcal{X} = \bigcup_{i=1}^m \mathcal{X}_i$.

$k_1, \dots, k_m \in \mathbb{Z}^+$.

$\gamma \in \mathbb{R}^+$: A guess of the optimum fair diversity.

Output: k_i points in \mathcal{X}_i for $i \in [m]$.

- 1: Define boolean function $f(p'_1, \dots, p'_m, j)$ that evaluates to 1 iff there exists $P \subseteq C_j$ with $\text{div}(P) \geq \gamma$ and $|P \cap \mathcal{X}_i| = p'_i$ for all $i \in [m]$.
- 2: Initialize $H \in \{0, 1\}^{(k_1+1) \times \dots \times (k_m+1) \times t}$ to 0.
- 3: Set $H[p'_1, \dots, p'_m, 1] \leftarrow f(p'_1, \dots, p'_m, 1)$.
- 4: **for** $j = 1$ to t **do**
- 5: For $k'_i \in \{0, \dots, k_i\} \forall i \in [m]$, update the entries in H as:

$$H[k'_1, \dots, k'_m, j] \leftarrow \bigvee_{\substack{p'_i \leq k'_i \\ \forall i \in [m]}} H[k'_1 - p'_1, \dots, k'_m - p'_m, j-1] f(p'_1, \dots, p'_m, j).$$

6: **end for**

7: **return** $H[k_1, k_2, \dots, k_m, n]$.

Algorithm 3 FAIR-EUCLIDEAN: A bi-criteria algorithm

Input: $\mathcal{X} = \bigcup_{i=1}^m \mathcal{X}_i$: points in \mathbb{R}^D with doubling dimension λ .

$k_1, \dots, k_m \in \mathbb{Z}^+$.

$\mathcal{T} = \bigcup_{i=1}^m T_i$: A coreset for fair Max-Min.

$\gamma \in \mathbb{R}^+$: A guess of the optimum fair diversity.

$\epsilon \in [0, 1]$: approximation error parameter.

Output: k_i points in \mathcal{X}_i for $i \in [m]$.

- 1: $\widehat{T}_i \leftarrow$ a maximal prefix of points in T_i such that $\text{div}(\widehat{T}_i) \geq \epsilon\gamma/4$.
 - 2: $p \leftarrow$ a point selected uniformly at random from $[0, W]^D$, where $W = 2mD\gamma/\epsilon$.
 - 3: Construct axis-aligned cubes $\mathcal{C} = \{C_1, C_2, \dots, C_t\}$ of side length W using p as one of the corners.
 - 4: In each cube C_i , remove all the points that are within a distance of $\gamma/2$ from one of the boundaries.
 - 5: **return** $\mathcal{S} \leftarrow \text{FAIR-DP}(C_1, \dots, C_t, (1-\epsilon)k_1, \dots, (1-\epsilon)k_m, \gamma)$.
-

$P \subseteq C_j$ with $\text{div}(P) \geq \gamma$ and p'_i points from each group i ; and, among the first $j-1$ clusters, there is a set with $k'_1 - p'_1, k'_2 - p'_2, \dots, k'_i - p'_i, \dots, k'_m - p'_m$ points from each group $i \in [m]$ that are at least γ far apart (the function f in FAIR-DP (Algorithm 2) evaluates where there is such a set P). We enumerate over all possible subsets of C_j to identify the subset P .

See FAIR-DP (Algorithm 2) for additional details and implementation. For simplicity, the algorithm is written to only determine whether it is possible to pick a subset with diversity γ subject to the required fairness constraints. Similar to FAIR-LINE, the algorithm can be easily extended to construct a subset of points for every non-zero entry in H by storing a pointer to the choice we made.

Theorem 6. *If $\gamma = \ell^*$, then, FAIR-DP (Algorithm 2) returns a set \mathcal{S} that satisfies $\text{div}(\mathcal{S}) \geq \ell^*$ and $|\mathcal{S} \cap \mathcal{X}_i| \geq k_i \forall i \in [m]$ and has a running time of $O(\prod_{i=1}^m (k_i + 1)^{2Rt})$ where $R = \max\{|C_1|, |C_2|, \dots, |C_t|\}$.*

Now, we describe a $1+\epsilon$ approximation algorithm for Euclidean metrics called FAIR-EUCLIDEAN that achieves $1-\epsilon$ fairness.

Overview of Fair-Euclidean. As part of the input, we construct a $(1+\epsilon)$ -coreset $\mathcal{T} = \bigcup_{i=1}^m T_i$

of size $O((8/\epsilon)^\lambda km)$ using the CORESET algorithm described in Section 4.2. We further assume a guess γ for the optimal diversity score ℓ^* . Note that the coreset \mathcal{T} is only constructed once and used for different guesses of ℓ^* .

For a fixed guess γ , for every group $i \in [m]$, we select a maximal prefix of points $\hat{T}_i \subset T_i$ that are at least $\epsilon\gamma/4$ far apart and define $\hat{\mathcal{T}} = \bigcup_{i=1}^m \hat{T}_i$.

Our main idea is to partition $\hat{\mathcal{T}}$ and obtain a collection of sets $\mathcal{C} = \{C_1, C_2, \dots, C_t\}$ separated by at least γ distance; thus any pair of points $x \in C_i$ and $y \in C_j$, $\forall i, j$ such that $i \neq j$, is separated by distance at least γ . Then, we use FAIR-DP on these sets C_1, C_2, \dots, C_t , and recover a solution \mathcal{S} with diversity γ .

To this end, we partition the points in $\hat{\mathcal{T}}$ into axis-aligned *cubes* $\mathcal{C} = \{C_1, C_2, \dots, C_t\}$ of length $W = 2mD\gamma/\epsilon$ as follows: we select a point p uniformly at random from $[0, W]^D$. Using p as one of the corners, we form axis-aligned cubes of length W until every point in \mathcal{X} is in one of the cubes. Then, from every cube $C_i \forall i \in [t]$ we remove every point of $\hat{\mathcal{T}}$ that is within a distance of $\gamma/2$ from one of its boundaries. Notice that any point that was not removed from a cube is at least γ far apart from any other point in a different cube. However, points within the same cube can be arbitrarily close. It is now easy to see that we can use FAIR-DP (Algorithm 2) on \mathcal{C} to retrieve a sufficient number of points from each group in $[m]$.

In the analysis below, we show that with probability at least $1/2$, we are able to find at least $(1 - \epsilon)k_i$ points from each group $i \in [m]$ that are all γ far apart.

Analysis. Let $\mathcal{S}^* = \bigcup_{i=1}^m \mathcal{S}_i^* \subset \mathcal{T}$ denote the optimal solution for FAIR MAX-MIN on the coreset $\mathcal{T} = \bigcup_{i=1}^m T_i$ with $\text{div}(\mathcal{S}^*) \geq \ell^*/(1 + \epsilon)$. Note that the optimal solution in \mathcal{T} is some subset in $\hat{\mathcal{T}}$ (see Theorem 5).

As a first step, we bound the number of optimal points \mathcal{S}_i^* from a group $i \in [m]$ that are removed by FAIR-EUCLIDEAN because they are within a distance of $\gamma/2$ from one of the boundaries of a cube.

Lemma 4. $\Pr[\forall i \in [m] : |\bigcup_{j \in [t]} C_j \cap \mathcal{S}_i^*| \geq (1 - \epsilon)k_i] \geq 1/2$.

Proof. Let $T'_i = \bigcup_{j \in [t]} C_j \cap \hat{T}_i$ be the remaining points in \hat{T}_i that are not close to the boundaries of any cube. Note that the FAIR-EUCLIDEAN algorithm *succeeds* if after the removal step there are least $(1 - \epsilon)k_i$ optimal points from each group i that can be selected by FAIR-DP at the final step of the algorithm while it *fails* otherwise. Below, we show that the probability it *succeeds* is at least $1/2$.

We compute the probability that a point $q \in \hat{T}_i$ is not removed by FAIR-EUCLIDEAN, i.e., $q \in T'_i$. It is removed if it lies within a distance of $\gamma/2$ from its boundaries in each dimension. Therefore, for q to remain in T'_i , the point p selected randomly from $[0, W]^D$ must not fall within a range of total length γ , in each dimension, which gives us:

$$\Pr[q \notin T'_i] = 1 - \Pr[q \in T'_i] = 1 - \left(\frac{W - \gamma}{W} \right)^D \leq \gamma D / W = \epsilon / 2m .$$

Fix a specific optimum solution. Define A_i be the number of points removed from this solution that are in group i . By Markov's inequality, $\Pr[A_i \geq k_i\epsilon] \leq \frac{\mathbf{E}[A_i]}{k_i\epsilon} \leq \frac{k_i\epsilon/(2m)}{k_i\epsilon} = \frac{1}{2m}$.

Taking union bound over all groups $i \in [m]$, we can bound the probability of discarding more than $k_i\epsilon$ points from some group i , $\Pr[\exists i \in [m] : A_i \geq k_i\epsilon] \leq \sum_{i=1}^m \Pr[A_i \geq k_i\epsilon] < 1/2$, and the lemma follows. \square

FAIR-DP depends exponentially on the number of points remaining in each cube (see Theorem 6). Now, we show that the total number of points remaining in each cube does not depend on n or k , and depends only on m, D, ϵ .

Lemma 5. $|C_j| \leq m \cdot (8mD^{3/2}/\epsilon^2)^\lambda$ for all $j \in [t]$.

We showed that for a fixed guess γ , the *success* probability of FAIR-EUCLIDEAN is $\geq 1/2$. Note that the only randomization used by FAIR-EUCLIDEAN is in selecting p . In order to increase the probability of success to $1 - \delta$ for some small $\delta \in (0, 1)$, we repeatedly select η points uniformly at random from $[0, W]^D$ as the corners. For each corner, we obtain a solution using FAIR-EUCLIDEAN, and we output the solution with the biggest diversity which also satisfies the fairness constraints with a loss of $(1 - \epsilon)$ multiplicative factor. The value of $\eta = \log(1/\delta)$ is selected such that the failure probability is $(1/2)^\eta < \delta$.

Note that the construction of the coreset \mathcal{T} allows us to reduce the number of guesses on ℓ^* from $O(n^2)$ to $O(|\mathcal{T}|^2) = O((8/\epsilon)^{2\lambda} k^2 m^2)$, which are all the pairwise distances in \mathcal{T} . Further, the number of clusters (i.e., cubes) in FAIR-EUCLIDEAN is upper bounded by the size of the coreset \mathcal{T} , which does not depend on n . The running time of FAIR-EUCLIDEAN depends on the running time to construct the coreset, which is $O((8/\epsilon)^\lambda kmn)$, and the running time of FAIR-DP (Algorithm 2) on the cubes \mathcal{C} . Since the number of points in each cube is $O(m \cdot (8mD^{3/2}/\epsilon^2)^\lambda)$, we conclude with the following theorem:

Theorem 7. *If $\gamma \geq \ell^*/(1 + \epsilon)$, FAIR-EUCLIDEAN Algorithm returns a set \mathcal{S} such that $\text{div}(\mathcal{S}) \geq \ell^*/(1 + \epsilon)$ and $|\mathcal{S} \cap \mathcal{X}_i| \geq k_i(1 - \epsilon) \forall i \in [m]$ with probability at least $1 - \delta$. For constant D, m , the running time is $O(nk + \text{poly}(1/\epsilon, k, \log(1/\delta)))$.*

In Appendix C, we give the exact running time with all the parameters. We can observe that the running time depends doubly exponentially on the doubling dimension, which is not uncommon for diversity maximization in doubling dimension metrics [16, 18].

5 Scalable Implementations

5.1 Data Stream Algorithms

In this section, we present single pass data stream algorithms that obtain the same approximation guarantees as that of sequential algorithms, while using low space. Missing details from this section are presented in Appendix D.

5.1.1 Extending Previous Algorithms

First, we describe an algorithm called τ -GMM that processes points sequentially, and includes a point in the solution if it is at least the threshold τ apart from every point in the current solution set. The set of points returned by τ -GMM are all separated by a distance of at least τ . If $m = 1$, then, we can set $\tau = \ell^*/2$ (using guessing for ℓ^*), and τ -GMM returns a solution set that is also a 2-approximation for the FAIR MAX-MIN problem [26]. τ -GMM allows us to extend it to data streaming setting, unlike the GMM algorithm which requires identifying the maximum distance point in each iteration.

Using τ -GMM with $\tau = \ell^*/2$, we can obtain a 5-coreset for general metrics [45], and $(1 + \epsilon)$ -coreset for Euclidean metrics (Section 4.2). Then, on the coreset, we use the randomized rounding algorithm from Section 3.2 and return the solution. This approach gives us the following guarantees:

Corollary 1. *There is a $O(\epsilon^{-1} km \log n)$ -space data stream algorithm that returns a $30(1 + \epsilon)$ -approximation with $(1 - \epsilon)$ -fairness for general metrics. For Euclidean metrics, there is a $O((8/\epsilon)^\lambda km \epsilon^{-1} \log n)$ space data stream algorithm that returns a $1 + \epsilon$ -approximation with $(1 - \epsilon)$ -fairness where λ is the doubling dimension of $\mathcal{X} \subset \mathbb{R}^D$.*

5.1.2 Improved Result for $m = 2$

In [45], the authors describe an algorithm called FAIR-SWAP which returns a 4-approximation to the FAIR MAX-MIN problem when the number of groups is $m = 2$. The algorithm can be directly extended to a 2-pass streaming algorithm using $O(k)$ space with the same 4-approximation guarantee. Building upon their work, and using new ideas we obtain a *single pass* algorithm FAIR-STREAM-2GROUPS which uses $O(k)$ space, and obtains 4-approximation to the FAIR MAX-MIN problem.

The algorithm maintains 3 sets S, S_1, S_2 using τ -GMM for all of them. In S , we include points in a group-agnostic way (similar to FAIR-SWAP) ignoring the fairness constraints. In S_1 , we include points only of group 1, and in S_2 , we include points only of group 2. By setting $\tau = \ell^*/2$ we maintain the sets S, S_1 and S_2 such that all points are at least $\ell^*/2$ distance apart in every one of them.

Without loss of generality, suppose \mathcal{X}_1 satisfies $|S \cap \mathcal{X}_1| < k_1$. Our algorithm proceeds by identifying $k_1 - |S \cap \mathcal{X}_1|$ additional points from S_1 denoted by Z_1 by running τ -GMM with $\tau = \ell^*/4$. This ensures that the final set of points from group 1, i.e., $(S \cap \mathcal{X}_1) \cup Z_1$ are $\ell^*/4$ apart. By discarding the nearest neighbors of newly added points (i.e., Z_1), in $S \cap \mathcal{X}_2$, we argue that our algorithm obtains a 4-approximation. We obtain the following guarantees:

Theorem 8. *There is a one-pass streaming algorithm that returns a $4(1 + \epsilon)$ -approximation for FAIR MAX-MIN problem using $O(k\epsilon^{-1} \log n)$ space.*

5.2 Composable Coresets

In this section, we design *composable* coresets for FAIR MAX-MIN. We assume the points \mathcal{X} are partitioned into L disjoint sets. We discuss an algorithm for constructing $(1 + \epsilon)$ -composable coresets for Euclidean metrics, and discuss extensions. Missing details are presented in Appendix E.

5.2.1 Constructing $(1 + \epsilon)$ -composable coresets

We assume the universe of points \mathcal{X} is partitioned into a collection of L disjoint sets $\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_L$. As in Section 4.2, we define an $\epsilon' > 0$ value such that $(1 - \epsilon') = 1/(1 + \epsilon)$.

We generalize the approach for constructing the coreset \mathcal{T} as follows: let \mathcal{Y}_j^i denote the points of group i present in \mathcal{Y}_j for $i \in [m]$ and $j \in [L]$. Then on each partition j and group i , we run GMM to retrieve a diverse set T_j^i with $O((4/\epsilon')^\lambda k)$, or equivalently $O((8/\epsilon)^\lambda k)$ points since $\epsilon' \geq \epsilon/2$. The coreset \mathcal{T} is defined as:

1. For $j \in [L]$, construct T_j : $T_j \leftarrow \bigcup_{i=1}^m T_j^i$, where $T_j^i \leftarrow \text{GMM}(\mathcal{Y}_j^i, (4/\epsilon')^\lambda k)$
2. $\mathcal{T} \leftarrow \bigcup_{j=1}^L T_j$

We obtain the following theorem:

Theorem 9. *\mathcal{T} is a $(1 + \epsilon)$ -composable coreset for fair Max-Min diversification of size $O((8/\epsilon)^\lambda kmL)$ in metrics of doubling dimension λ that can be obtained in $O((8/\epsilon)^\lambda kmnL)$ time.*

For general metrics, using a similar approach, we obtain a 5-composable coreset by extending a recent construction of 5-coreset for the sequential setting [45]. We also discuss two-pass distributed algorithms for constructing α -composable coresets for Euclidean ($\alpha = 1 + \epsilon$) and general metrics ($\alpha = 5$).

6 Conclusion

In this paper, we presented new approximation algorithms that substantially improve upon currently known results for the FAIR MAX-MIN problem both in general and Euclidean metric spaces. There are several interesting directions for future work, including obtaining a **2-approximation for the problem in general metrics or improving the hardness result.**

Another direction is to generalize the fairness constraints to arbitrary matroid constraints (the fairness constraints considered in this paper can be expressed via the special case of a partition matroid). While there are results known for related diversity maximization problems under matroid constraints [2, 11, 14], to the best of our knowledge, there are currently no results for Max-Min diversification.

Acknowledgements

This work was supported by the NSF under grants CCF-1934846, CCF-1908849, CCF-1637536, CCF-1763423, IIS-1943971, and an Adobe Research Grant.

References

- [1] Sofiane Abbar, Sihem Amer-Yahia, Piotr Indyk, Sepideh Mahabadi, and Kasturi R Varadarajan. Diverse near neighbor problem. In *Proceedings of the twenty-ninth annual symposium on Computational geometry*, pages 207–214, 2013.
- [2] Zeinab Abbassi, Vahab S. Mirrokni, and Mayur Thakur. Diversity maximization under matroid constraints. In *KDD '13*, pages 32–40, 2013.
- [3] P. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Geometric approximation via coresets. 2007.
- [4] Pankaj K. Agarwal, Graham Cormode, Zengfeng Huang, Jeff Phillips, Zhewei Wei, and Ke Yi. Mergeable summaries. In *PODS '12*, page 23–34, 2012.
- [5] Pankaj K. Agarwal, Stavros Sintos, and Alex Steiger. Efficient indexes for diverse top-k range queries. In *PODS '20*, page 213–227, 2020.
- [6] Sepideh Aghamolaei, Majid Farhadi, and Hamid Zarrabi-Zadeh. Diversity maximization via composable coresets. In *CCCG*, 2015.
- [7] Albert Angel and Nick Koudas. Efficient diversity-aware search. In *SIGMOD '11*, page 781–792, 2011.
- [8] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, May 1998.
- [9] Patrice Assouad. Plongements lipschitziens dans \mathbb{R}^n . *Bulletin de la Société Mathématique de France*, 111:429–448, 1983.
- [10] Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. *Advances in Neural Information Processing Systems*, 32:4954–4965, 2019.
- [11] Aditya Bhaskara, Mehrdad Ghadiri, Vahab Mirrokni, and Ola Svensson. Linear relaxations for finding diverse elements in metric spaces. In *NIPS'16*, page 4105–4113, 2016.

- [12] Michele Borassi, Alessandro Epasto, Silvio Lattanzi, Sergei Vassilvitskii, and Morteza Zadimoghaddam. Better sliding window algorithms to maximize subadditive and diversity objectives. In *PODS '19*, page 254–268, 2019.
- [13] Allan Borodin, Aadhar Jain, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions, and dynamic updates. *ACM Trans. Algorithms*, 2017.
- [14] Allan Borodin, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *PODS '12*, pages 155–166, 2012.
- [15] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98*, page 335–336, 1998.
- [16] Matteo Ceccarello, Andrea Pietracaprina, and Geppino Pucci. Fast coresot-based diversity maximization under matroid constraints. In *WSDM '18*, pages 81–89, 2018.
- [17] Matteo Ceccarello, Andrea Pietracaprina, and Geppino Pucci. A general coresot-based approach to diversity maximization under matroid constraints. *ACM Trans. Knowl. Discov. Data*, 2020.
- [18] Matteo Ceccarello, Andrea Pietracaprina, Geppino Pucci, and Eli Upfal. Mapreduce and streaming algorithms for diversity maximization in metric spaces of bounded doubling dimension. *Proc. VLDB Endow.*, page 469–480, 2017.
- [19] Elisa Celis, Vijay Keswani, Damian Straszak, Amit Deshpande, Tarun Kathuria, and Nisheeth Vishnoi. Fair and diverse DPP-based data summarization. In *ICML '2018*, pages 716–725, 2018.
- [20] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. Ranking with fairness constraints. In *ICALP*, 2017.
- [21] Alfonso Cevallos, Friedrich Eisenbrand, and Rico Zenklusen. Local search for max-sum diversification. In *SODA '17*, page 130–142, 2017.
- [22] Barun Chandra and Magnús M Halldórsson. Approximation algorithms for dispersion problems. *J. Algorithms*, pages 438–465, 2001.
- [23] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *NIPS*, 2017.
- [24] Ashish Chiplunkar, Sagar Kale, and Sivaramakrishnan Natarajan Ramamoorthy. How to solve fair k-center in massive data models. In *ICML 2020*, pages 1877–1886, 2020.
- [25] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.
- [26] Graham Cormode, S. Muthukrishnan, and Wei Zhuang. Conquering the divide: Continuous clustering of distributed data streams. In *ICDE 2007*, pages 1036–1045, 2007.
- [27] Ting Deng and Wenfei Fan. On the complexity of query result diversification. *ACM Transactions on Database Systems (TODS)*, 39(2):1–46, 2014.
- [28] M. Drosou and E. Pitoura. Diverse set selection over dynamic data. *IEEE Transactions on Knowledge and Data Engineering*, 26(5):1102–1116, 2014.

- [29] Marina Drosou, H.V. Jagadish, Evaggelia Pitoura, and Julia Stoyanovich. Diversity in big data: A review. *Big Data*, 5:73–84, 2017.
- [30] Marina Drosou and Evaggelia Pitoura. Search result diversification. *SIGMOD Rec.*, (1):41–47, 2010.
- [31] Marina Drosou and Evaggelia Pitoura. Disc diversity: Result diversification based on dissimilarity and coverage. *Proc. VLDB Endow.*, 6(1):13–24, November 2012.
- [32] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *ITCS '12*, pages 214–226, 2012.
- [33] Marwa El Halabi, Slobodan Mitrović, Ashkan Norouzi-Fard, Jakab Tardos, and Jakub M Tarnawski. Fairness in streaming submodular maximization: Algorithms and hardness. In *NeurIPS 2020*, volume 33, pages 13609–13622, 2020.
- [34] Erhan Erkut. The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48 – 60, 1990.
- [35] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. Fairness testing: Testing software for discrimination. In *ESEC/FSE '17*, pages 498–510, 2017.
- [36] Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *WWW '09*, page 381–390, 2009.
- [37] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985.
- [38] Sudipto Guha. Tight results for clustering and summarizing data streams. In *ICDT '09*, page 268–275, 2009.
- [39] Anupam Gupta, Robert Krauthgamer, and James R Lee. Bounded geometries, fractals, and low-distortion embeddings. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 534–543. IEEE, 2003.
- [40] Refael Hassin, Shlomi Rubinstein, and Arie Tamir. Approximation algorithms for maximum dispersion. *Oper. Res. Lett.*, 21(3):133–137, October 1997.
- [41] Piotr Indyk, Sepideh Mahabadi, Mohammad Mahdian, and Vahab S. Mirrokni. Composable core-sets for diversity and coverage maximization. In *PODS '14*, page 100–108, 2014.
- [42] Matthew Jones, Huy Nguyen, and Thy Nguyen. Fair k-centers via maximum matching. In *ICML 2020*, pages 4940–4949, 2020.
- [43] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In *ICML '19*, volume 97, pages 3448–3457, 09–15 Jun 2019.
- [44] Michael J. Kubly. Programming models for facility dispersion: The p-dispersion and maximum dispersion problems. *Geographical Analysis*, 19(4):315–329, 1987.
- [45] Zafeiria Moumoulidou, Andrew McGregor, and Alexandra Meliou. Diverse Data Selection under Fairness Constraints. In *ICDT 2021*, pages 13:1–13:25, 2021.

- [46] Lu Qin, Jeffrey Xu Yu, and Lijun Chang. Diversifying top-k results. *Proc. VLDB Endow.*, 5(11):1124–1135, July 2012.
- [47] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. Heuristic and special case algorithms for dispersion problems. *Oper. Res.*, 42(2):299–310, April 1994.
- [48] Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.
- [49] Julia Stoyanovich, Ke Yang, and H. V. Jagadish. Online set selection with fairness and diversity constraints. In *EDBT*, 2018.
- [50] Arie Tamir. Obnoxious facility location on graphs. *SIAM J. Discrete Math.*, 4:550–567, 11 1991.
- [51] Marcos R. Vieira, Humberto L. Razente, Maria C. N. Barioni, Marios Hadjieleftheriou, Divesh Srivastava, Caetano Traina, and Vassilis J. Tsotras. On query result diversification. In *ICDE 2011*, pages 1163–1174, 2011.
- [52] Yanhao Wang, Francesco Fabbri, and Michael Mathioudakis. Fair and representative subset selection from data streams. In *WWW 2021*, page 1340–1350, 2021.
- [53] Yue Wang, Alexandra Meliou, and Gerome Miklau. Rc-index: Diversifying answers to range queries. *Proc. VLDB Endow.*, 11(7):773–786, 2018.
- [54] Ke Yang, Vasilis Gkatzelis, and Julia Stoyanovich. Balanced ranking with diversity constraints. In *IJCAI’19*, pages 6035–6042, 2019.
- [55] Ke Yang and Julia Stoyanovich. Measuring fairness in ranked outputs. In *SSDBM ’17*, 2017.
- [56] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa*ir: A fair top-k ranking algorithm. In *CIKM ’17*, pages 1569–1578, 2017.

Appendix

A $(m + 1)$ -Approximation with Perfect Fairness

We give the pseudocode for FAIR-GREEDY-FLOW (Algorithm 4) described in Section 3.3 and analyze its approximation guarantees. Let \mathcal{C} denote the set of clusters obtained when the input γ (a guess for the optimal diversity value) to Algorithm 4 satisfies, $\ell^*/(1 + \epsilon) < \gamma \leq \ell^*$ for some $\epsilon > 0$. Let $\mathcal{S}^* = \{y_1, y_2, \dots, y_k\}$ denote the set of points selected by an optimal solution.

Recall that by construction of a cluster $D \in \mathcal{C}$, any point $p \in \mathcal{X}$ from some group i is removed from further consideration only if: (1) it is less than $\frac{\gamma}{m+1}$ distance from some point in D , and (2) there is already a point $p' \in D$ of group i . Using Claim 1, we show that in every cluster $D \in \mathcal{C}$ there is at most one optimal point $y_i \in \mathcal{S}^*$, which is either included in D , or if y_i was removed, then, there is a point $y' \in D$ of the same group as y_i .

Notation. Define $f(y_i) \in D$ to be a point of the same group as y_i , and note that $f(y_i)$ could be y_i itself. Let $f(\mathcal{S}^*) = \{f(y_1), f(y_2), \dots, f(y_k)\}$.

To be more specific, assume an optimal point $y_i \in \mathcal{S}^*$ is less than $\frac{\gamma}{m+1}$ distance from some point in D . Let $f(y_i)$ represent a point in D of the same group as y_i , and note that $f(y_i)$ could be y_i itself if y_i was not removed while constructing D ; otherwise $f(y_i) = y'$. Then, we show that there cannot be a pair of $f(y_i), f(y_j)$ points in the same cluster $D \in \mathcal{C}$ for any pair of optimal points $y_i, y_j \in \mathcal{S}^*$. Consequently, we can argue that there are at least k_i points from each group i in the formed clusters \mathcal{C} , some of which could be the optimal points themselves, and are all in *different* clusters. This will let us argue it is possible to find a flow of size k using the reduction to max-flow.

Claim 1. Let $y_i \in \mathcal{S}^*$ denote an optimal point that is less than $\frac{\gamma}{m+1}$ distance from some point in some cluster $D \in \mathcal{C}$. We have: $|\bigcup_{p \in D} \mathbf{B}(p, \frac{\gamma}{m+1}) \cap f(\mathcal{S}^*)| \leq 1$ for every cluster $D \in \mathcal{C}$.

Proof. For the sake of contradiction, assume there is pair of points

$$f(y_1), f(y_2) \in \bigcup_{p \in D} \mathbf{B}(p, \frac{\gamma}{m+1}) \cap f(\mathcal{S}^*)$$

for a pair of optimal points $y_1, y_2 \in \mathcal{S}^*$. From the definition of optimality, we have: $d(y_1, y_2) \geq \ell^* \geq \gamma$.

Let $p^{y_1} \in D$ and $p^{y_2} \in D$ be the points from which y_1 (and y_2 respectively), is $< \frac{\gamma}{m+1}$ distance apart. Note that it is not necessarily the case that $p^{y_1} = f(y_1)$ (or $p^{y_2} = f(y_2)$). As D contains at most m points, note that the largest chain in D has length of $m - 1$ and thus by triangle inequality and construction of D , $d(p^{y_1}, p^{y_2}) \leq \frac{(m-1)\gamma}{m+1}$. Once again using triangle inequality, we have:

$$\begin{aligned} d(y_1, y_2) &\leq d(y_1, p^{y_1}) + d(y_2, p^{y_2}) + d(p^{y_1}, p^{y_2}) \\ &< \frac{\gamma}{m+1} + \frac{\gamma}{m+1} + \frac{(m-1)\gamma}{m+1} < \gamma. \end{aligned}$$

This is a contradiction. Consequently, it follows that there is at most one optimal point $y_i \in \mathcal{S}^*$ in every cluster $D \in \mathcal{C}$, and it is either included in D , in which case $f(y_i) = y_i$, or there is another point y' from the same group as y_i present in D (i.e., $f(y_i) = y'$). \square

Now, we argue that the set of all points in \mathcal{C} contains a subset of points that obtains an $m + 1$ -approximation factor.

Claim 2. There exists a subset $S \subseteq \bigcup_{D \in \mathcal{C}} D$ such that

Algorithm 4 FAIR-GREEDY-FLOW

Input: $\mathcal{X} = \bigcup_{i=1}^m \mathcal{X}_i$: Universe of available elements.
 $k_1, \dots, k_m \in \mathbb{Z}^+$.
 $\gamma \in \mathbb{R}^+$: A guess of the optimum fair diversity.

Output: k_i points in \mathcal{X}_i for $i \in [m]$.

- 1: $\mathcal{R} \leftarrow \mathcal{X}$ denote the set of remaining elements.
- 2: $\mathcal{C} \leftarrow \emptyset$ denote a collection of subsets of points (called clusters).
- 3: **while** $|\mathcal{R}| > 0$ (**and**) $|\mathcal{C}| \leq km$ **do**
- 4: $D \leftarrow \emptyset$ denote the current cluster, and $D_{\text{col}} \leftarrow \emptyset$ denote the groups of points in cluster D .
- 5: **while** an element $p \in \mathcal{R} \cap \mathcal{X}_i$ for some $i \in \{1, 2, \dots, m\} \setminus D_{\text{col}}$ exists **do**
- 6: **if** $|D| = 0$ (or) $d(p, x) < \frac{\gamma}{m+1}$ for some $x \in D$ **then**
- 7: $D \leftarrow D \cup \{p\}$ and $D_{\text{col}} \leftarrow D_{\text{col}} \cup \{i\}$.
- 8: **end if**
- 9: **end while**
- 10: $\mathcal{R} \leftarrow \mathcal{R} \setminus \bigcup_{p \in D} \mathbf{B}(p, \frac{\gamma}{m+1})$.
- 11: $\mathcal{C} \leftarrow \mathcal{C} \cup \{D\}$.
- 12: $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{X}_i \forall i \in [m]$ if $|\{D \mid D \in \mathcal{C} \text{ and } D \cap \mathcal{X}_i \neq \emptyset\}| \geq k$.
- 13: **end while**
- ▷Construct flow graph :
- 14: Let $\mathcal{C} = \{D_1, D_2, \dots, D_t\}$.
- 15: Construct directed graph $G = (V, E)$ where

$$\begin{aligned} V &= \{a, u_1, \dots, u_m, v_1, \dots, v_t, b\} \\ E &= \{(a, u_i) \text{ with capacity } k_i : i \in [m]\} \\ &\quad \cup \{(v_j, b) \text{ with capacity } 1 : j \in [t]\} \\ &\quad \cup \{(u_i, v_j) \text{ with capacity } 1 : |\mathcal{X}_i \cap D_j| \geq 1\} \end{aligned}$$
- 16: Set $\mathcal{S} \leftarrow \emptyset$. Compute maximum a - b flow in G using Ford-Fulkerson algorithm [25].
- 17: **if** flow size $< k = \sum_i k_i$ **then return** \emptyset ▷Abort
- 18: **else** ▷max flow is k
- 19: $\forall (u_i, v_j)$ with flow equal to 1, add the point in D_j with group i to \mathcal{S} .
- 20: **end if**
- 21: **return** \mathcal{S} .

(i) $|S \cap D| \leq 1$ for every $D \in \mathcal{C}$ and $|S \cap \mathcal{X}_j| = k_j$ for every $j \in [m]$.

(ii) For every pair of points $s_1, s_2 \in S$, $d(s_1, s_2) \geq \frac{\gamma}{m+1}$.

Proof. (i) We show that in the formed clusters \mathcal{C} , it possible to select k_j points from each group j by selecting at most one point from each cluster $D \in \mathcal{C}$.

Following the analysis in [45], call a group j *non-critical* if there are k points in the formulated set of clusters \mathcal{C} . Otherwise, call group j *critical*. Note that if a group j is *critical*, FAIR-GREEDY-FLOW processed all the points in \mathcal{X}_j . For non-critical groups, recall that the remaining points are discarded once we find k points.

For a critical group j , from Claim 1 and the fact that all points were processed while constructing the clusters, we know that every cluster $D \in \mathcal{C}$ satisfies $|\bigcup_{p \in D} \mathbf{B}(p, \frac{\gamma}{m+1}) \cap f(\mathcal{S}_j^*)| \leq 1$. Therefore, it follows that for a critical group j there are k_j distinct clusters among the formed clusters that contain the points in $f(\mathcal{S}_j^*)$. We add the points in $f(\mathcal{S}_j^*)$ to S .

Note that for all *non-critical* groups, the selected points are all in different clusters by construction. Thus, regardless of how pick the points for the different groups across the clusters, by the time we process some non-critical group j there will be at least $k - \sum_{i: \text{processed groups}} k_i \geq k_j$ clusters, each of which contain one point from group j . These k_j points can be added to S by picking only one point from each cluster.

Thus, it follows that we can pick k_j points for each group $j \in [m]$ by picking at most one point from each cluster $D \in \mathcal{C}$.

(ii) Consider two points $s_1, s_2 \in S$. As $|S \cap D| \leq 1$ for every $D \in \mathcal{C}$, we have that s_1 and s_2 belong to different clusters, say D^{s_1} and D^{s_2} . Without loss of generality suppose the cluster D^{s_1} is formed before D^{s_2} . From Algorithm 4, we know that all the points in $\mathbf{B}(s_1, \frac{\gamma}{m+1})$ are removed, and s_2 is chosen from the remaining points. Therefore, $d(s_1, s_2) \geq \frac{\gamma}{m+1}$. \square

Claim 3. *There is a valid flow of size k in the directed graph G as described in FAIR-GREEDY-FLOW (Algorithm 4).*

Proof. Let \mathcal{C} be the clustering obtained by FAIR-GREEDY-FLOW. Consider the sum of the capacities of all edges incident on the source node a in the graph G . It is given by $\sum_{i \in [m]} k_i = k$. Therefore, the maximum flow in G is at most k . We argue that we can construct a valid flow of size k from \mathcal{C} .

Let $S \subseteq \bigcup_{D \in \mathcal{C}} D$ be a subset satisfying Claim 2. For every point $e \in S$ such that $e \in D_j \cap \mathcal{X}_i$ for some $D_j \in \mathcal{C}, i \in [m]$, we add a flow of value 1 on the edges (u_i, v_j) , (v_j, b) and (a, u_i) . As $|S \cap \mathcal{X}_i| = k_i$ (Claim 2 (i)), we have a flow of value k_i on every edge of the form (a, u_i) for $i \in [m]$. As $|S \cap D_j| \leq 1$ (Claim 2 (i)), the capacity constraints are always satisfied for the edges incident on node v_j for every $D_j \in \mathcal{C}$. As $|S \cap \mathcal{X}_i| = k_i$ (Claim 2 (i)), the capacity constraints are satisfied for the edges incident on node u_i for all $i \in [m]$. Therefore, there is a valid flow achieving the maximum flow $\sum_{i \in [m]} k_i = k$ in the graph G . \square

Claim 4. *Given a flow f of size k in G , the set S obtained by FAIR-GREEDY-FLOW (Algorithm 4) satisfies the fairness constraints, and for every two points $s_1, s_2 \in S$, we have $d(s_1, s_2) \geq \frac{\gamma}{m+1}$.*

Proof. As the sum of the capacities of all edges incident on node a in the graph G is $\sum_{i \in [m]} k_i = k$, we can conclude that an edge incident on node u_i has a flow of k_i in f . From conservation of flow, at every node u_i , there are k_i edges of the form (u_i, v_j) with flow 1. We include a node from D_j of group i in S if the flow on the edge (u_i, v_j) is 1. Therefore, we have included k_i nodes of group i in S for every $i \in [m]$ satisfying the fairness constraints.

Consider $s_1, s_2 \in S$ belonging to clusters D^{s_1} and D^{s_2} respectively. From the construction, every edge (v_j, b) of capacity 1 ensures that only one point from a cluster in \mathcal{C} is picked. Therefore, we have D^{s_1} and D^{s_2} are distinct clusters. Without loss of generality let the cluster D^{s_1} be formed before D^{s_2} . From Algorithm 4, we know that all the points in $\mathbf{B}(s_1, \frac{\gamma}{m+1})$ are removed, and s_2 is chosen from the remaining points. Therefore, $d(s_1, s_2) \geq \frac{\gamma}{m+1}$. \square

Theorem 10 (Theorem 2 restated). *FAIR-GREEDY-FLOW (Algorithm 4) returns a $(m+1)(1+\epsilon)$ -approximation and achieves perfect fairness for the FAIR MAX-MIN problem using a running time of $O(nkm^3\epsilon^{-1} \log n)$.*

Proof. Suppose the guess $\gamma \geq \ell^*/(1+\epsilon)$ is obtained using binary search. Let \mathcal{C} denote the clusters obtained by Algorithm 4 using the guess γ . From Claim 3, we have that there is a valid flow of size k in G . After solving the maximum flow problem on the graph G , let f be the flow. Using Claim 4, we can obtain a set S from f that satisfies fairness constraints, and for every $s_1, s_2 \in S$, we have: $d(s_1, s_2) \geq \frac{\gamma}{m+1} = \frac{\ell^*}{(m+1)(1+\epsilon)}$. Hence, S obtains an approximation factor of $(m+1)(1+\epsilon)$.

Running Time. Each cluster $D \in \mathcal{C}$ contains at most m points. For including a point $p \in D$, we iterate over all remaining points \mathcal{R} that are within a distance of $\frac{\gamma}{m+1}$, and of a group different from the points already included in D . This step requires a running time of $O(|\mathcal{R}| \cdot m \cdot m) = O(nm^2)$. We remove all points of group $i \in [m]$ from \mathcal{R} , if the total number of clusters containing a point from the group is at least k . This ensures that there are at most km clusters that are possible.

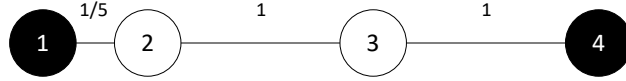
Therefore, the total running time for constructing \mathcal{C} is $O(nm^2 \cdot km) = O(nkm^3)$.

We can observe that the number of edges in G are $O(|\mathcal{C}|m) = O(km^2)$. Using Ford-Fulkerson, we can obtain the maximum flow of value k . This step requires a running time of $O(k^2m^2)$. For identifying the correct guess γ , we use binary searching which results in a multiplicative factor of $O(\log n/\epsilon)$ to the total running time (see Section 3.1 for a discussion). Combining all the running times, gives us the lemma. \square

A.1 Tight Example for $3m - 1$ approximation algorithm in [45]

Below we show a tight example for FAIR-FLOW in [45] and show how FAIR-GREEDY-FLOW yields a better approximation.

A tight example for Fair-Flow. Suppose $k = 3$ and we have to select one white and two black points. Here, edges represent the distance across two points, e.g., $d(p_1, p_2) = 1/5$. Note that the optimal solution in this example is the set of points $\{p_1, p_3, p_4\}$ with diversity score equal to 1.



The FAIR-FLOW algorithm in [45], for a guess $\gamma = 1$, for the black group selects both points since they are at least $d_1 = \frac{m\gamma}{3m-1} = 2/5$ far apart from each other. Similarly for the white group. Now because there is no pair of points with distance strictly less than $d_2 = \frac{\gamma}{3m-1} = 1/5$, FAIR-FLOW constructs four connected components (each with a point). As a result, the points $\{p_1, p_2, p_4\}$ will be selected by the max-flow algorithm and we obtain a set with diversity score equal to $1/5$. Note that for this example, FAIR-GREEDY-FLOW returns the set $\{p_1, p_3, p_4\}$ as p_1 and p_2 are less than $1/3$ distance apart. These two points will be in the same cluster and at most one of them can be picked; thus, we guarantee an approximation ratio of 3.

B Hardness of Approximation

Theorem 11 (Theorem 3 restated). *Unless $P = NP$, there is no polynomial time algorithm for the FAIR MAX-MIN problem that obtains an α -approximation factor for diversity score, and a β -approximation for the fairness constraints, i.e., the set \mathcal{S} returned by the algorithm satisfies $|\mathcal{S} \cap \mathcal{X}_i| \geq k_i/\beta \ \forall i \in [m]$, for some constants $\alpha < 2$ and $\beta > 0$.*

Proof. We present a reduction from GAP-CLIQUE $_\rho$, where $\rho = \beta$. For every vertex of the graph G , we create a new point, and set of points is denoted by \mathcal{X} . For every edge (u, v) in G , we set $d(u, v) := 2$. For all other pairs of vertices, we set the distances as 1. Every vertex is assigned the same color, and the corresponding fairness constraint is $|\mathcal{S} \cap \mathcal{X}| \geq k$, where \mathcal{S} is the set of points whose diversity we are trying to maximize in FAIR MAX-MIN.

Suppose there is a polynomial time algorithm that returns a set \mathcal{S} , obtains an α -approximation for the diversity score, and a β -approximation for the fairness constraints. We first consider the ‘YES’ instance in GAP-CLIQUE $_\beta$, i.e., we assume there is a clique of size k in G . This implies $\ell^* = 2$. As $\alpha < 2$, we have that the set \mathcal{S} returned has a diversity score $\geq \ell^*/\alpha > 1$. Therefore, \mathcal{S} is a clique in G as all other pairwise distances are 1 (from construction). As \mathcal{S} is a β -approximation for the fairness constraint, we have that $|\mathcal{S}| \geq k/\beta$. Let us now consider the ‘No’ instance, i.e., there is no clique of size $\geq k/\beta$ in G . Therefore, $|\mathcal{S} \cap \mathcal{X}| < k/\beta$, as $|\mathcal{S} \cap \mathcal{X}|$ is upper bounded by the maximum clique size in G . From the above arguments, we have that using our algorithm, we can distinguish the ‘YES’ and ‘NO’ instances of GAP-CLIQUE $_\beta$, which is not possible unless $P = NP$ [8]. Hence, the theorem. \square

C Euclidean Metrics

C.1 Exact Computation in One Dimension

Theorem 12 (Theorem 4 restated). *There is an algorithm that solves the FAIR MAX-MIN problem exactly when the points can be embedded on a line and requires a running time of $O(n^4 \prod_{i=1}^m (k_i + 1))$.*

Proof. We use FAIR-LINE to identify the exact solution. We observe that any optimal solution can be expressed as a subset of the first j points for some $j \in [n]$. From the construction, if the guess $\gamma \leq \ell^*$ there will always be at least k_i points from group i for all $i \in [m]$ that are all γ far apart. Therefore, since the dynamic programming approach finds all the subsets with k_i points per group i for all $j \in [n]$, at least one of the $H[k_1, k_2, \dots, k_m, j]$ entries will be equal to 1 as required. As discussed previously, we can backtrack and construct the solution set.

Running Time. For a fixed guess γ , we need to compute $\prod_{i=1}^m (k_i + 1)$ entries for every point, as every k'_i for $i \in [m]$ takes at most $k_i + 1$ values. To compute an entry $H[\cdot, \cdot, \dots, \cdot, j]$ using FAIR-LINE (Algorithm 1), we need to retrieve $O(n)$ distances to find point j' that is at least γ far apart from point j . Thus, the total running is equal to $O(n^2 \prod_{i=1}^m (k_i + 1))$ since there are $O(n \prod_{i=1}^m (k_i + 1))$ entries in H and the computational cost to fill each entry is $O(n)$. As there are $O(n^2)$ distance values the guess γ can take, the total running time is $O(n^4 \prod_{i=1}^m (k_i + 1))$. \square

C.2 $(1 + \epsilon)$ Approx with $(1 - \epsilon)$ Fairness

Theorem 13 (Theorem 6 restated). *If $\gamma = \ell^*$, then, FAIR-DP (Algorithm 2) returns a set \mathcal{S} that satisfies $\text{div}(\mathcal{S}) \geq \ell^*$ and $|\mathcal{S} \cap \mathcal{X}_i| \geq k_i \forall i \in [m]$ and has a running time of $O(\prod_{i=1}^m (k_i + 1)^2 2^{Rt})$ where $R = \max\{|C_1|, |C_2|, \dots, |C_t|\}$.*

Proof. As $\gamma = \ell^*$, the optimal set of points satisfy the fairness constraints. From the construction in FAIR-DP, we will return a set \mathcal{S} that has diversity ℓ^* , and achieves perfect fairness.

Running Time. Consider a value $j \in [t]$. There are $\prod_{i=1}^m (k_i + 1)$ entries in the table H corresponding to this value of j . For every $k'_i \in \{0, 1, \dots, k_i\}$ and every subset $R \subseteq C_j$ where $|R \cap \mathcal{X}_i| = p'_i \forall i \in [m]$, we check if there is a valid subset of points satisfying fairness constraints using the condition mentioned in FAIR-DP. Since there at most $\prod_{i=1}^m (k_i + 1)$ ways to enumerate the p'_i values (because $p'_i \leq k'_i$), the total time to compute entries corresponding to this j value is $O(\prod_{i=1}^m (k_i + 1)^2 2^{Rt})$. Therefore, to compute all the entries in H we need $O(\prod_{i=1}^m (k_i + 1)^2 2^{Rt})$ time. \square

Lemma 6 (Lemma 5 restated). *For every $j \in [t]$, $|C_j| \leq m \cdot (8mD^{3/2}/\epsilon^2)^\lambda$ where λ is doubling dimension of \mathbb{R}^D .*

Proof. Consider all points in C_j that belong to group i , i.e., $C_j \cap \hat{T}_i$. From the construction of $\hat{T}_i \subseteq \mathcal{T}$, we have that every pair of points of the same group is separated by a distance at least $\epsilon\gamma/4$. Therefore, each point can be represented by a ball of radius $\epsilon\gamma/8$, and we want to count the maximum number of non-overlapping balls that can be packed inside the cube C_j . Observe that the length of the diagonal of C_j is $W\sqrt{D}$, and the cube lies entirely in the ball of radius $W\sqrt{D}/2$ with center at the middle of the diagonal. We call this *cube ball*. As Euclidean metrics are doubling metrics, we can cover the *cube ball* with overlapping balls of radius $\epsilon\gamma/8$ and the number of the balls required is $\left(\frac{W\sqrt{D}/2}{\epsilon\gamma/8}\right)^\lambda$, where $\lambda = O(D)$ is the doubling dimension of \mathbb{R}^D .

We can observe that the total volume occupied by the overlapping balls is at least the volume occupied by the non-overlapping balls corresponding to the points and having the same radius. Therefore, we can upper bound the number of points using the total number of non-overlapping

balls used to cover the *cube ball*. As there are m groups, we have that the total number of the points in C_j is: $|C_j| \leq m \cdot \left(\frac{W\sqrt{D}/2}{\epsilon\gamma/8}\right)^\lambda = m \cdot (8mD^{3/2}/\epsilon^2)^\lambda$. \square

Theorem 14 (Theorem 7 restated). *If $\gamma \geq \ell^*/(1 + \epsilon)$, FAIR-EUCLIDEAN Algorithm returns a set \mathcal{S} such that $\text{div}(\mathcal{S}) \geq \ell^*/(1 + \epsilon)$ and $|\mathcal{S} \cap \mathcal{X}_i| \geq k_i(1 - \epsilon) \forall i \in [m]$ with probability at least $1 - \delta$. The running time of the algorithm is:*

$$O(n \cdot (8/\epsilon)^\lambda km + \prod_{i=1}^m (k_i + 1)^{2^{m(8mD^{3/2}/\epsilon^2)^\lambda}} (8/\epsilon)^\lambda km \log |\mathcal{T}| \log(1/\delta)).$$

Proof. The running time of FAIR-EUCLIDEAN (Algorithm 3) depends on: (1) the running time of constructing the coreset \mathcal{T} which is $O(nkm(8/\epsilon)^\lambda)$, where λ is the doubling dimension, and (2) the running time of FAIR-DP (Algorithm 2) on the clusters for every guess γ .

From Theorem 6, we know that FAIR-DP has a running time of $O(\prod_{i=1}^m (k_i + 1)^{2^R t})$, where t is the number of clusters and R is the maximum size across all t clusters. We upper bound the number of clusters by the coreset size. So, $t = O((8/\epsilon)^\lambda km)$. From Lemma 5, we have $R = O(m(8mD^{3/2}/\epsilon^2)^\lambda)$. Combining all the above, the final running time is:

$$O((8/\epsilon)^\lambda kmn + \log |\mathcal{T}| \log(1/\delta) \prod_{i=1}^m (k_i + 1)^{2^{m(8mD^{3/2}/\epsilon^2)^\lambda}} (8/\epsilon)^\lambda km).$$

\square

D Scalable Algorithms: Data Stream Algorithms

In this section, we present 1-pass data stream algorithms that obtain good approximation guarantees for the FAIR MAX-MIN problem. First, we present a variant called τ -GMM, of the GMM algorithm from Section 4.2. Instead of finding the point that is farthest away, we use a threshold τ to identify the next point to be included in the set of diverse points. The threshold τ is set to $\gamma/2$, for a guess γ of the optimal diversity score ℓ^* . We argue that this threshold variant of GMM algorithm (called τ -GMM) also obtains the same 2-approximation guarantee.

Using τ -GMM, we give several data stream algorithms for the FAIR MAX-MIN problem. In Section D.1, we give an algorithm that obtains a 30-approximation using $O(km)$ space for general metrics. In Section D.2, we give an algorithm that obtains a $(1 + \epsilon)$ -approximation using $O(km(8/\epsilon)^\lambda)$ space, where $\lambda = O(D)$ for D -dimensional Euclidean metrics. Finally, for the special case of two groups, i.e., $m = 2$, we give an algorithm that obtains a $4(1 + \epsilon)$ -approximation using $O(k\epsilon^{-1} \log n)$ space for general metrics.

Overview of τ -GMM. We initialize the set of diverse points denoted by \mathcal{S} with some arbitrary point in \mathcal{X} . In each iteration, the algorithm selects an arbitrary point p among the available points, denoted by \mathcal{R} , and includes it in \mathcal{S} , if the distance between the selected point p and all other points in \mathcal{S} is greater than $\tau = \gamma/2$. The point p is made unavailable by removing it from \mathcal{R} for subsequent iterations, irrespective of its inclusion in \mathcal{S} . Finally, the set \mathcal{S} is returned.

As defined in [47], FAIR MAX-MIN when $m = 1$ is also called MAX-MIN diversification. Using the theorem below, when our guess γ is ℓ^* , we show that τ -GMM obtains a 2-approximation for MAX-MIN diversification problem:

Algorithm 5 Algorithm τ -GMM

Input: \mathcal{X} : Universe of available points.
 $\tau \in \mathbb{R}^+$: a threshold on distance.
 $k \in \mathbb{Z}^+$.
 \mathcal{I} : initialization set of points.

Output: $\mathcal{S} \subseteq \mathcal{X}$ of size k .

```
1: Let  $\mathcal{R} \leftarrow \mathcal{X}$  denote set of remaining points.
2: If  $\mathcal{I} \neq \emptyset$ , initialize  $\mathcal{S} \leftarrow \mathcal{I}$ , otherwise  $\mathcal{S} \leftarrow$  arbitrarily chosen point in  $\mathcal{X}$ .
3: while  $|\mathcal{S}| < k$  and  $\mathcal{R} \neq \emptyset$  do
4:   Let  $p \in \mathcal{R}$  be an arbitrary point.
5:   if  $\min_{v \in \mathcal{S}} d(p, v) \geq \tau$  then
6:      $\mathcal{S} \leftarrow \mathcal{S} \cup \{p\}$ .
7:   end if
8:    $\mathcal{R} \leftarrow \mathcal{R} \setminus \{p\}$ .
9: end while
10: return  $\mathcal{S}$ 
```

Theorem 15. *If $\gamma = \ell^*$, the algorithm τ -GMM with $\tau = \gamma/2$ returns a 2-approximation for the MAX-MIN diversification problem.*

Proof. From the construction of set \mathcal{S} , $d(p, q) \geq \tau = \gamma/2 = \ell^*/2$ for every pair $p, q \in \mathcal{S}$, we have the approximation guarantee. Now, we will argue that $|\mathcal{S}| = k$. Let $\mathcal{S}^* = \{y_1, y_2, \dots, y_k\}$ denote the set of points that obtains optimal diversity score, i.e., $\text{div}(\mathcal{S}^*) = \ell^*$. We claim that for every point $p \in \mathcal{S}$, there exists at most one point $f(p) \in \mathcal{S}^*$ such that $d(p, f(p)) < \ell^*/2$. For the sake of contradiction suppose there are two points $y, y' \in \mathcal{S}^* \cap \mathbf{B}(p, \ell^*/2)$ such that $d(p, y) < \ell^*/2$ and $d(p, y') < \ell^*/2$. From triangle inequality, we have $d(y, y') < \ell^*$, which is not true because $d(y, y') \geq \text{div}(\mathcal{S}^*) = \ell^*$. For every point $p \in \mathcal{S}$, consider $\mathbf{B}(p, \ell^*/2)$. As $d(p, q) \geq \gamma/2 = \ell^*/2$ for every pair $p, q \in \mathcal{S}$, and $|\mathbf{B}(p, \ell^*/2) \cap \mathcal{S}^*| \leq 1$, there are $|\mathcal{S}^*|$ balls containing exactly one point from \mathcal{S}^* and centered at a point from \mathcal{S} . Therefore, $|\mathcal{S}| \geq k$ and τ -GMM terminates when $|\mathcal{S}| = k$. Hence, the theorem. \square

Now, we describe an algorithm τ -GMM-STREAM that is based on τ -GMM and is a data streaming algorithm.

Overview of τ -GMM-Stream. The algorithm runs τ -GMM for each group $i \in [m]$ separately using the same threshold τ . Therefore, when a point $p \in \mathcal{X}_j$ in the stream is processed, τ -GMM-STREAM checks if it can include it in the solution \mathcal{S}_j corresponding to group $j \in [m]$ using the threshold τ . The algorithm returns $\bigcup_{j \in [m]} \mathcal{S}_j$ containing \tilde{k}_j points of group j , for all $j \in [m]$.

D.1 Data streaming algorithm for General metrics

In this section, we describe a data stream algorithm called FAIR-STREAM-GEN that obtains a 30-approximation for the FAIR MAX-MIN problem.

In [45], the authors give a simple coresets construction for general metrics, by taking a union of the outputs of GMM algorithm run on each group separately. They argue that by selecting at most k points using GMM from each group, the resulting set of $O(km)$ points in total, contains a 5-approximation for the FAIR MAX-MIN problem. Building on this, we give a data stream algorithm that first constructs the coresets using τ -GMM-STREAM, and then, we use RANDOMIZED-ROUNDING from Section 3.1 to return a 6-approximation on the coresets.

Algorithm 6 Algorithm τ -GMM-STREAM

Input: \mathcal{X} : Universe of available points.

$\tau \in \mathbb{R}^+$: a threshold on distance.

$\tilde{k}_1, \dots, \tilde{k}_m \in \mathbb{Z}^+$.

Output: $\mathcal{S} \subseteq \mathcal{X}$ of size $\tilde{k}_1 + \tilde{k}_2 + \dots + \tilde{k}_m$.

- 1: Initialize $\mathcal{S}_i \leftarrow \emptyset \forall i \in [m]$.
 - 2: **while** processing point $p \in \mathcal{X}_j$ for some $j \in [m]$ **do**
 - 3: **if** $|\mathcal{S}_j| < \tilde{k}_j$ and $\min_{v \in \mathcal{S}_j} d(p, v) \geq \tau$ **then**
 - 4: $\mathcal{S}_j \leftarrow \mathcal{S}_j \cup \{p\}$.
 - 5: **end if**
 - 6: **end while**
 - 7: **return** $\mathcal{S} \leftarrow \bigcup_{i=1}^m \mathcal{S}_i$.
-

Algorithm 7 Algorithm FAIR-STREAM-GEN

Input: \mathcal{X} : Universe of available points.

$k_1, \dots, k_m \in \mathbb{Z}^+$.

$d_{\min}^{\text{lb}}, d_{\max}^{\text{ub}}$: lower bound for minimum and upper bound
for maximum pairwise distances in \mathcal{X} .

Output: $\mathcal{S} \subseteq \mathcal{X}$.

- 1: **for** each of the guesses $\gamma \in \Gamma := \{d_{\min}^{\text{lb}}, (1+\epsilon)d_{\min}^{\text{lb}}, (1+\epsilon)^2 d_{\min}^{\text{lb}}, \dots, d_{\max}^{\text{ub}}\}$, run in parallel **do**
 - 2: Set $\tilde{k}_i = k \forall i \in [m]$.
 - 3: $T_i \leftarrow \tau\text{-GMM-STREAM}(\mathcal{X}_i, 2\gamma/5, \tilde{k}_1, \tilde{k}_2, \dots, \tilde{k}_m) \forall i \in [m]$.
 - 4: $\mathcal{T}_\gamma \leftarrow \bigcup_{i=1}^m T_i$.
 - 5: **end for**
 - 6: Let x^* be the optimal LP solution to FAIR MAX-MIN LP for the points $\bigcup_{\gamma \in \Gamma} \mathcal{T}_\gamma$.
 - 7: **return** $\mathcal{S} \leftarrow \text{RANDOMIZED-ROUNDING}(\bigcup_{\gamma \in \Gamma} \mathcal{T}_\gamma, x^*)$.
-

Let $d_{\min} = \min_{p,q \in \mathcal{X}} d(p, q)$ and $d_{\max} = \max_{p,q \in \mathcal{X}} d(p, q)$. We assume we are given a lower bound d_{\min}^{lb} for d_{\min} , and an upper bound d_{\max}^{ub} for d_{\max} , following similar assumptions for fair k -center clustering in the data streaming setting [24]. We use a parameter γ to guess optimal diversity score ℓ^* .

Overview of Fair-Stream-Gen. We identify the coreset using τ -GMM-STREAM for a guess γ , we set $\tau = 2\gamma/5$ and $\tilde{k}_j = k \forall j \in [m]$. The coreset \mathcal{T}_γ consists of $O(\tilde{k}_j)$ points of group j , all separated by a distance of $2\gamma/5$. We use geometric guessing in the range $\Gamma := \{d_{\min}^{\text{lb}}, (1+\epsilon)d_{\min}^{\text{lb}}, (1+\epsilon)^2 d_{\min}^{\text{lb}}, \dots, d_{\max}^{\text{ub}}\}$ using the parameter γ and run all these $O(\frac{1}{\epsilon} \log(d_{\max}^{\text{ub}}/d_{\min}^{\text{lb}}))$ guesses in parallel.

Finally, we construct FAIR MAX-MIN LP on the set of coreset points for all the guesses, given by, $\bigcup_{\gamma \in \Gamma} \mathcal{T}_\gamma$. The fractional solution x^* of this LP is then used to construct an integral solution using RANDOMIZED-ROUNDING from Section 3.1.

Combining the 5-approximation of coreset from [45] and the guarantees of RANDOMIZED-ROUNDING, we obtain the following theorem:

Theorem 16. *The algorithm FAIR-STREAM-GEN returns a \mathcal{S} that obtains a $30(1+\epsilon)$ -approximation for the diversity score and $1-\epsilon$ -fairness for the FAIR MAX-MIN problem using $O(km \frac{1}{\epsilon} \log(d_{\max}^{\text{ub}}/d_{\min}^{\text{lb}}))$ space. Here, $d_{\max}^{\text{ub}} \geq \max_{p,q \in \mathcal{X}} d(p, q)$ and $d_{\min}^{\text{lb}} \leq \min_{p,q \in \mathcal{X}} d(p, q)$ are given upper and lower bound estimates.*

Proof. If $\gamma = \ell^*/(1+\epsilon)$, following the same proof of Theorem 4 in [45], we can argue that \mathcal{T}_γ is a $5(1+\epsilon)$ -approximate coreset. Therefore, $\text{div}(\mathcal{T}_\gamma) \geq \ell^*/5(1+\epsilon)$. Combining this with the guarantees

Algorithm 8 Algorithm FAIR-STREAM-EUCLIDEAN

Input: \mathcal{X} : Universe of available points.

$k_1, \dots, k_m \in \mathbb{Z}^+$.

$d_{\min}^{\text{lb}}, d_{\max}^{\text{ub}}$: lower bound for minimum and upper bound
for maximum pairwise distances in \mathcal{X} .

Output: $\mathcal{S} \subseteq \mathcal{X}$.

- 1: **for** each of the guesses $\gamma \in \Gamma := \{d_{\min}^{\text{lb}}, (1+\epsilon)d_{\min}^{\text{lb}}, (1+\epsilon)^2d_{\min}^{\text{lb}}, \dots, d_{\max}^{\text{ub}}\}$, run in parallel **do**
 - 2: Set $\tilde{k}_i = k(8/\epsilon)^\lambda \forall i \in [m]$ where λ is doubling dimension of \mathbb{R}^D .
 - 3: $T_i \leftarrow \tau\text{-GMM-STREAM}(\mathcal{X}_i, \epsilon\gamma/4, \tilde{k}_1, \tilde{k}_2, \dots, \tilde{k}_m) \forall i \in [m]$.
 - 4: $\mathcal{T}_\gamma \leftarrow \bigcup_{i=1}^m T_i$.
 - 5: **end for**
 - 6: Let x^* be the optimal LP solution to FAIR MAX-MIN LP for the points $\bigcup_{\gamma \in \Gamma} \mathcal{T}_\gamma$.
 - 7: **return** $\mathcal{S} \leftarrow \text{RANDOMIZED-ROUNDING}(\bigcup_{\gamma \in \Gamma} \mathcal{T}_\gamma, x^*)$.
-

of Theorem 1, we get a $30(1+\epsilon)$ -approximation and $1-\epsilon$ fairness. \square

D.2 Data streaming algorithm for Euclidean metrics

In this section, we describe a data stream algorithm called FAIR-STREAM-EUCLIDEAN that obtains a 6-approximation for FAIR MAX-MIN. In Section 4.2, we gave an algorithm to construct a $(1+\epsilon)$ -approximate coreset for Euclidean metrics, by taking a union of the outputs of GMM algorithm run on each group. Using τ -GMM-STREAM, we obtain a $(1+\epsilon)$ -approximate coreset. Similar to Section D.1, we use RANDOMIZED-ROUNDING from Section 3.1 to return a 6-approximation on the coreset.

Let $d_{\min} = \min_{p,q \in \mathcal{X}} d(p,q)$ and $d_{\max} = \max_{p,q \in \mathcal{X}} d(p,q)$. We assume we are given a lower bound d_{\min}^{lb} for d_{\min} , and an upper bound d_{\max}^{ub} for d_{\max} , following similar assumptions for fair k -center clustering in the data streaming setting [24]. We use a parameter γ to guess optimal diversity score ℓ^* .

Overview of Fair-Stream-Euclidean. We identify the coreset using τ -GMM-STREAM for a guess γ , we set $\tau = \epsilon\gamma/4$ and $\tilde{k}_j = k(8/\epsilon)^\lambda \forall j \in [m]$, where λ is doubling dimension of \mathbb{R}^D . The coreset \mathcal{T}_γ consists of \tilde{k}_j points of group j , all separated by a distance of $\epsilon\gamma/4$. We use geometric guessing in the range $\Gamma := \{d_{\min}^{\text{lb}}, (1+\epsilon)d_{\min}^{\text{lb}}, (1+\epsilon)^2d_{\min}^{\text{lb}}, \dots, d_{\max}^{\text{ub}}\}$ using the parameter γ and run all these $O(\frac{1}{\epsilon} \log(d_{\max}^{\text{ub}}/d_{\min}^{\text{lb}}))$ guesses in parallel.

On the coreset obtained, we use FAIR-DP, and the pairwise distances of points in the coreset are the new guesses for ℓ^* . We obtain the following guarantees:

Theorem 17. *The algorithm FAIR-STREAM-EUCLIDEAN returns a \mathcal{S} that obtains a $(1+\epsilon)$ -approximation for the diversity score of FAIR MAX-MIN problem using $O(km(8/\epsilon)^{1+\lambda} \log(d_{\max}^{\text{ub}}/d_{\min}^{\text{lb}}))$ space where λ is doubling dimension of \mathbb{R}^D and $d_{\max}^{\text{ub}} \geq \max_{p,q \in \mathcal{X}} d(p,q)$, $d_{\min}^{\text{lb}} \leq \min_{p,q \in \mathcal{X}} d(p,q)$ are given input upper and lower bound estimates.*

Proof. If $\gamma = \ell^*/(1+\epsilon)$, following the same proof as Theorem 5, we can argue that \mathcal{T}_γ is a $(1+\epsilon)$ -approximate coreset. Therefore, $\text{div}(\mathcal{T}_\gamma) \geq \ell^*/(1+\epsilon)$. Combining this with the guarantees of Theorem 6, we get a $(1+\epsilon)$ approximation. \square

D.3 Data streaming algorithm when $m = 2$

In this section, we describe a data streaming algorithm FAIR-STREAM-2GROUPS which obtains a 4-approximation to the FAIR MAX-MIN problem when the number of groups $m = 2$. In [45],

the authors describe an algorithm called FAIR-SWAP which returns a 4-approximation to the FAIR MAX-MIN problem when the number of groups is $m = 2$. The algorithm can be extended to a 2-pass streaming algorithm using $O(k)$ space with the same 4-approximation guarantee. Therefore, our algorithm essentially reduces the number of passes with the same guarantees for the space used. First, we describe the algorithm FAIR-SWAP from [45] briefly and then present an overview of our Algorithm 9.

Let $d_{\min} = \min_{p,q \in \mathcal{X}} d(p, q)$ and $d_{\max} = \max_{p,q \in \mathcal{X}} d(p, q)$. We assume we are given a lower bound d_{\min}^{lb} for d_{\min} , and an upper bound d_{\max}^{ub} for d_{\max} , following similar assumptions for fair k -center clustering in the data streaming setting [24]. We use a parameter γ to guess optimal diversity score ℓ^* .

The FAIR-SWAP algorithm described in [45] uses GMM to retrieve a set \mathcal{S} of k points in a group-agnostic way by ignoring the fairness constraints. Thus, it is possible that one group is under-represented (i.e., $|\mathcal{S} \cap \mathcal{X}_i| < k_i$) and the other is over-represented (i.e., $|\mathcal{S} \cap \mathcal{X}_i| > k_i$). So, the algorithm first identifies the under-represented group i for some $i \in \{1, 2\}$. Then, it uses GMM only on \mathcal{X}_i , and obtains $\tilde{\mathcal{S}}_i$, called the *swap-set* containing the remaining $k_i - |\mathcal{S} \cap \mathcal{X}_i|$ points. Finally for every point in $\tilde{\mathcal{S}}_i$ it removes the nearest neighbor in the over-represented group.

Overview of Algorithm Fair-Stream-2Groups. We first note that for a guess $\gamma = \ell^*$ we can *simulate* the behavior of GMM using τ -GMM algorithm given in the Appendix. In order to extend FAIR-SWAP to a single pass data streaming algorithm, we need to know the under-represented group. However, we can only determine it towards the end of running the τ -GMM algorithm. We overcome this by simultaneously treating either of the two groups as under-represented whenever a new point in the stream is selected by the τ -GMM algorithm. In order to do that, we maintain 3 sets S, S_1, S_2 using τ -GMM for all of them. In S , we include points in a group-agnostic way (similar to FAIR-SWAP) ignoring the fairness constraints. In S_1 , we include points only of group 1, and in S_2 , we include points only of group 2. By setting $\tau = \gamma/2$ we maintain the sets S, S_1 and S_2 such that all points are at least $\gamma/2$ distance apart in every one of them.

Without loss of generality, suppose \mathcal{X}_1 is the under-represented group. So, $|\mathcal{S} \cap \mathcal{X}_1| < k_1$. Our algorithm proceeds by identifying $k_1 - |\mathcal{S} \cap \mathcal{X}_1|$ new points from S_1 by running τ -GMM with the initialization set of $\mathcal{S} \cap \mathcal{X}_1$ and $\tau = \gamma/4$. This ensures that the final set of points from group 1 that are returned are $\gamma/4$ apart. By discarding the nearest neighbors of group 2 from newly added points, we argue that our algorithm obtains a 4-approximation.

We obtain the following guarantees for FAIR-STREAM-2GROUPS:

Theorem 18 (Theorem 8 restated). *There is a data streaming algorithm that obtains a $4(1 + \epsilon)$ -approximation for FAIR MAX-MIN problem using $O(k\epsilon^{-1} \log(d_{\max}^{\text{ub}}/d_{\min}^{\text{lb}}))$ space, where $d_{\max}^{\text{ub}} \geq \max_{p,q \in \mathcal{X}} d(p, q)$ and $d_{\min}^{\text{lb}} \leq \min_{p,q \in \mathcal{X}} d(p, q)$ are given input upper and lower bound distance estimates.*

Proof. Consider the optimal solution $\mathcal{S}^* = \{y_1, y_2, \dots, y_k\}$ where $k = k_1 + k_2$. Consider set S_1 after the stream has ended. We can observe that for every point $p \in S_1$, $|\mathbf{B}(p, \ell^*/2) \cap \mathcal{S}^*| \leq 1$. Therefore, $|S_1| = k_1$. Similarly, we can argue that $|S| = k$ and $|S_2| = k_2$.

Let $T_i = \mathcal{S} \cap \mathcal{X}_i$ for $i \in \{1, 2\}$ and $u \in \{1, 2\}$ denote the under-represented group, i.e., $|T_u| < k_u$ and o denote the over-represented group, i.e., $|T_o| > k_o$. This means that we need to find $k_u - |T_u|$ additional points to add to S to satisfy the fairness constraint for group u . First, we argue that S_u contains $k_u - |T_u|$ points that are at least $\gamma/4$ distance from all the points in T_u . As $\tau = \gamma/2$, we have $\text{div}(S_u) \geq \gamma/2$ and $\text{div}(T_u) \geq \gamma/2$. Consider balls of radius $\gamma/4$ centered around points in T_u , given by $\mathbf{B}(p, \gamma/4) \forall p \in T_u$. Using triangle inequality, we have that $|\mathbf{B}(p, \gamma/4) \cap S_u| \leq 1 \forall p \in T_u$, as otherwise, we would have two points in S_u with distance strictly less than $\gamma/2$. This implies

Algorithm 9 Algorithm FAIR-STREAM-2GROUPS: Data Stream Algorithm for $m = 2$.

Input: $\mathcal{X}_1, \mathcal{X}_2$: Universe of available points.

$k_1, k_2 \in \mathbb{Z}^+$.

$d_{\min}^{\text{lb}}, d_{\max}^{\text{ub}}$: lower bound for minimum and upper bound for maximum pairwise distances in \mathcal{X} .

Output: Set \mathcal{S} of k_1, k_2 points in $\mathcal{X}_1, \mathcal{X}_2$.

```

1: for each of the guesses  $\gamma \in \Gamma := \{d_{\min}^{\text{lb}}, (1 + \epsilon)d_{\min}^{\text{lb}}, (1 + \epsilon)^2 d_{\min}^{\text{lb}}, \dots, d_{\max}^{\text{ub}}\}$ , run in parallel do
2:   Initialize  $S \leftarrow \emptyset$ ,  $S_1 \leftarrow \emptyset$  and  $S_2 \leftarrow \emptyset$ .
3:   while processing point  $p \in \mathcal{X}_j$  from stream do
4:     if  $d(p, S) \geq \gamma/2$  and  $|S| < k$  then
5:        $S \leftarrow S \cup \{p\}$ .
6:     end if
7:     if  $j = 1$  and  $d(p, S_1) \geq \gamma/2$  and  $|S_1| < k_1$  then
8:        $S_1 \leftarrow S_1 \cup \{p\}$ .
9:     end if
10:    if  $j = 2$  and  $d(p, S_2) \geq \gamma/2$  and  $|S_2| < k_2$  then
11:       $S_2 \leftarrow S_2 \cup \{p\}$ .
12:    end if
13:  end while
14:   $T_j \leftarrow S \cap \mathcal{X}_j$  for  $j \in \{1, 2\}$ .
15:  Set  $u \leftarrow \arg \min_j |T_j| - k_i$  and  $o \leftarrow 3 - u$ .
16:   $E_u \leftarrow \tau\text{-GMM}(S_u, \gamma/4, k_u - |T_u|, T_u)$ 
17:   $R_o \leftarrow \{\arg \min_{q \in T_o} d(p, q) : p \in E_u \setminus T_u\}$ .
18:   $\mathcal{S}_\gamma \leftarrow E_u \cup (T_o \setminus R_o)$ .
19: end for
20: return  $\mathcal{S} \leftarrow \arg \max_{\mathcal{S}_\gamma: \gamma \in \Gamma} \text{div}(\mathcal{S}_\gamma)$ 

```

that there are at least $k_u - |T_u|$ points in S_u , denoted by \tilde{S}_u with distance at least $\gamma/4$ from points in T_u . As $\tilde{S}_u \subseteq S_u$, we have $\text{div}(\tilde{S}_u) \geq \gamma/2$. Therefore, using τ -GMM on the set S_u with $\tau = \gamma/4$ and initialized with points T_u , we obtain the set E_u containing k_u points such that $\text{div}(E_u) \geq \gamma/4$.

We remove the nearest neighbors denoted by the set R_o from \mathcal{X}_o of the newly included points $E_u \setminus T_u$. As $\text{div}(T_o) \geq \gamma/2$, we have for $\text{div}(R_o) \geq \gamma/2$. Using triangle inequality, we have that $|\mathbf{B}(p, \gamma/4) \cap R_o| \leq 1 \forall p \in E_u \setminus T_u$, as otherwise, we would have two points in R_o with distance strictly less than $\gamma/2$. Therefore, the points remaining given by $\mathcal{S}_\gamma = E_u \cup (T_o \setminus R_o)$ satisfy $\text{div}(\mathcal{S}_\gamma) \geq \gamma/4$. For the guess $\gamma \geq \ell^*/(1 + \epsilon)$, we have that: $\text{div}(\mathcal{S}_\gamma) \geq \frac{\ell^*}{4(1 + \epsilon)}$.

As the total number of points stored is $O(|S| + |S_1| + |S_2|) = O(k)$, the theorem follows. \square

E Scalable Algorithms: Distributed Systems & Composable Coresets

In this section we show the approach discussed in Section 5.2 yields an $(1 + \epsilon)$ -composable coreset \mathcal{T} for FAIR MAX-MIN problem. Our proof is similar to Theorem 5 and the main difference is in identifying *critical* and *non-critical* groups based on the properties of $T_j \forall j \in [m]$.

The analysis uses the *anti-cover* property, which every set selected by GMM satisfies. Let $S = \{s_1, s_2, \dots, s_k\} \subseteq \mathcal{X}$ be a set of size k selected by GMM. Then, we say S is an *r-net* for \mathcal{X} , or equivalently S satisfies the *anti-cover* property, if for $r = \min_{u \in \{S \setminus s_k\}} d(u, s_k) = \text{div}(S)$, the following properties hold: (1) (*separation*) for any $u, v \in S$, $d(u, v) \geq r$, and (2) (*coverage*) for any $v \in \mathcal{X} \setminus S$, $\min_{u \in S} d(u, v) \leq r$. We now prove the following result:

Theorem 19 (Theorem 9 restated). \mathcal{T} is a $(1 + \epsilon)$ -composable coreset for fair Max-Min diversification of size $O((8/\epsilon)^\lambda kmL)$ in metrics of doubling dimension λ which can be obtained in $O((8/\epsilon)^\lambda kmnL)$ time.

Proof. We show how by analyzing the properties of the T_j sets, we can categorize the groups into *critical* and *non-critical*. Then the ideas in Theorem 5 in the sequential setting apply here as well.

Let T_i be the set of points of group i present in \mathcal{T} , namely for a fixed value i : $T_i \leftarrow \bigcup_{j=1}^L T_j^i$. For any point $x \in \mathcal{X}_i$ for $i \in [m]$, define $f(x) = \min_{y \in T_i} d(x, y)$. By the *anti-cover* property of GMM, each $T_j^i \subset T_i$ set of points that belong to group i is an r_j^i -net for \mathcal{Y}_j^i , where $r_j^i = \text{div}(T_j^i)$. Then T_j^i satisfies the *separation* and *coverage* properties, thus the following hold:

1. For any pair of points $x, y \in T_j^i$, it holds that $d(x, y) \geq r_j^i$.
2. For any point $x \in \mathcal{Y}_j^i \setminus T_j^i$, it holds that $d(x, f(x)) \leq r_j^i$.

Now, following the analysis of Indyk et al. [41], define $r^i = \max_{j \in [L]} r_j^i$. Then by the coverage property it follows that $d(x, f(x)) \leq r^i$ for any point $x \in \mathcal{X}_i$ in group i for $i \in [m]$. Also notice that if $|T_j^i| < (4/\epsilon')^\lambda k$ for all $j \in [L]$, then r^i can be treated as zero since $d(x, f(x)) = 0$ for all points in \mathcal{X}_i . Further, since T_i is a superset of the T_j^i sets, it contains a set $T'_i \subseteq T_i$ with $\text{div}(T'_i) \geq r^i$. Next, notice that we can define the following cases for the value of r^i to see if a group is critical or not:

1. If $r^i \geq (\epsilon'/2)\ell^*$ group i is *non-critical*. Also notice there exists a set in T_i with at least $(4/\epsilon')^\lambda k$ points that are greater or equal than $(\epsilon'/2)\ell^*$ apart from each other.
2. If $r^i < (\epsilon'/2)\ell^*$ group i is *critical*. Also notice that for any point $x \in \mathcal{X}_i$ it holds that $d(x, f(x)) \leq r^i < (\epsilon'/2)\ell^*$.

Finally, using similar arguments as in the sequential setting in Theorem 5, we can prove that \mathcal{T} is a $(1 + \epsilon)$ -composable coreset. \square

We now discuss a simple two-round distributed algorithm for FAIR MAX-MIN that uses composable coresets.

A two-round distributed algorithm using composable coresets. We assume the data is partitioned into L processing sites. A similar approach based on coresets was proposed by [18] for the unconstrained Max-Min diversification problem. In the first round, each site $j \in [L]$ computes its local coreset T_j and sends it to the coordinator site. In the second round, at the coordinator site we use the best known approximation algorithm in the sequential setting or a brute force approach. Here, in the latter case we retrieve an $(1 + \epsilon)$ -approximate solution. Alternatively, we could use the linear programming approach discussed in Section 3. In that case, we get a $6(1 + \epsilon)$ -approximate solution for the FAIR MAX-MIN problem by only sacrificing at most ϵk_i points per group i . Similar arguments follow if we are in general metric spaces.