Diverse Data Selection under Fairness Constraints

Zafeiria Moumoulidou
College of Information and Computer Sciences, University of Massachusetts Amherst, MA, USA

Andrew McGregor
College of Information and Computer Sciences, University of Massachusetts Amherst, MA, USA

Alexandra Meliou
College of Information and Computer Sciences, University of Massachusetts Amherst, MA, USA

Abstract

Diversity is an important principle in data selection and summarization, facility location, and recommendation systems. Our work focuses on maximizing diversity in data selection, while offering fairness guarantees. In particular, we offer the first study that augments the Max-Min diversification objective with fairness constraints. More specifically, given a universe $\mathcal{U}$ of n elements that can be partitioned into m disjoint groups, we aim to retrieve a k-sized subset that maximizes the pairwise minimum distance within the set (*diversity*) and contains a pre-specified k_i number of elements from each group i (*fairness*). We show that this problem is NP-complete even in metric spaces, and we propose three novel algorithms, linear in n, that provide strong theoretical approximation guarantees for different values of m and k. Finally, we extend our algorithms and analysis to the case where groups can be overlapping.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases data selection, diversity maximization, fairness constraints, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.ICDT.2021.13

Funding This work was supported by the NSF under grants CCF-1934846, CCF-1908849, CCF-1637536, IIS-1453543, CCF-1763423, and IIS-1943971.

1 Introduction

Data is generated and collected from all aspects of human activity, in domains like commerce, medicine, and transportation, as well as scientific measurements, simulations, and environmental monitoring. However, while datasets grow large and are readily available, they are often down-sampled for various uses. This is often due to practical implications, e.g., analytics workflows may be designed, tested, and debugged over subsets of the data for efficiency reasons. Other times, machine learning applications use subsets of the data for training and testing, while applications that target human consumption, e.g., data exploration, can only display small parts of the data at a time, since human users can visually process limited information.

While data subset selection is very common, deriving *good* subsets is a non-trivial task. In this paper, we focus on two principles in data selection: *diversity* and *fairness*. Diversity and fairness are related but distinct concepts. Specifically, diversity seeks to maximize the dissimilarity of the items in a set. Intuitively, a diverse set of items selected from a dataset D represents more and different aspects of the information present in D. Prior work has suggested several diversity objectives [16, 30, 32, 43], typically defined in terms of an element-wise distance function over numerical attributes (e.g., geographic location, age). On the other hand, fairness aims to achieve some specified level of representation across different categories or groups, and is typically defined over categorical attributes (e.g., race, gender). While one could consider combining fairness and diversity into a single objective,

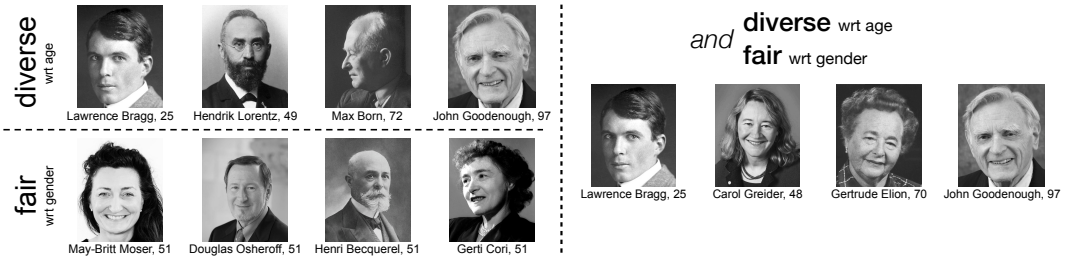


Figure 1 Three examples of selection of four items from a dataset of Nobel laureates. The first set on the left is diverse with respect to age; the second set is fair with respect to gender; the third set, on the right, is both diverse with respect to age and fair with respect to gender.

comparing numerical and categorical attributes is not straightforward, as it typically requires ad hoc decisions in discretizing numerical attributes or defining a distance function involving numeric and categorical attributes.

Figure 1 demonstrates an example of the principles of diversity and fairness in subset selection. Consider a web search query over a dataset of Nobel laureates. There are close to a thousand laureates, but the web search only serves a small number of results for human consumption. Figure 1 shows three examples of possible subsets of four items. The first subset optimizes the set's diversity with respect to the age of the laureates at the time of the award, but only contains male scientists. The second set achieves fair gender representation, but is not diverse with respect to age. The third set achieves both diversity and fairness. The concept of *fair* and *diverse* data selection is motivated by many real-world scenarios: *transportation equity* in conjunction with optimizing traditional objectives (e.g., geographic coverage) aims to design accessible transportation systems for historically disadvantaged groups [37]; formulating *teams* that represent various demographic groups while demonstrating "diversity of thought" is becoming an important hiring goal [21, 23, 31]; in *news* websites, a summary of dissimilar in context documents from different news channels minimizes redundancy and mitigates the risk of showing a polarized opinion [23].

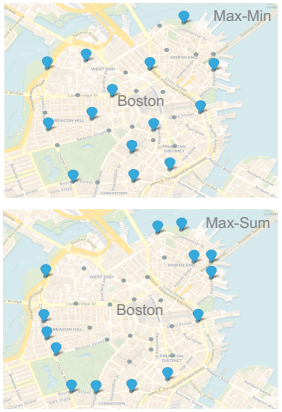
Our focus. In this paper, *our goal is to maximize diversity in data selection* with respect to numerical attributes, *while ensuring the satisfaction of fairness constraints* with respect to categorical ones. We focus on the Max-Min diversification model [23, 43, 46], which is among the most well-studied and frequently-used diversity models in the data management community. Max-Min diversification seeks to select a set of k items, such that the distance between any two items is maximized. We further express fairness as cardinality constraints: given m demographic groups, a set is *fair* if it contains a pre-specified integer number k_i of representatives from each group. This general form of cardinality constraints captures, among others, the common fairness objectives of *proportional* representation, where the sample preserves the demographic proportions of the general population, and *equal* representation, where all demographic groups are equally represented in the sample.[1] These fairness objectives have been widely studied in prior work [13, 14, 35, 45, 48, 49, 50], and fit naturally in problems of data selection where existing systems can exhibit bias with respect to sensitive attributes; e.g., a study showed that search engines tend to under-represent women in the result sets [34].

[1] Our fairness constraints are based on the definitions of *group fairness* and *statistical parity* [25]. Other definitions that focus on *individual or causal fairness* examine differences in treatment of individuals from different groups who are otherwise very similar, but these are not the focus of this work.

	Max-Min	Max-Sum
diversification	[30, 43, 46] $\frac{1}{2}$-approximation	
fair diversification (disjoint groups)	[this paper] $\frac{1}{4}$-approx. $(m = 2)$ $\frac{1}{3m-1}$-approx. $(m \geq 3)$ $\frac{1}{5}$-approx. $(m = O(1)$ and $k = o(\log n))$	[1, 9, 11] $(\frac{1}{2} - \epsilon)$-approx.
fair diversification (overlapping groups)	[this paper] $\frac{1}{4}$-approx. $(m = 2)$ $\frac{1}{3\binom{m}{\lfloor m/2 \rfloor}-1}$-approx. $(m \geq 3)$	N/A

n : # elements in the universe, m : # demographic groups, k : # elements in the data selection task

(a) Comparison with prior art.



(b) Max-Min vs Max-Sum.

Figure 2 (a) Contributions of this paper with respect to the prior art. Our work is the first to introduce fairness constraints to Max-Min diversification, and provides strong approximation results. We also contribute algorithms to the case of overlapping classes, which has not been addressed in prior work. (b) The department of transportation wants to place $k = 14$ new bike sharing stations in downtown Boston among $n = 30$ candidate locations. (Top): Max-Min selects locations that geographically *cover* downtown. (Bottom): Max-Sum selects locations on the outskirts of downtown.

We first study the problem of *fair Max-Min diversification* in the case of *non-overlapping* groups, and define the problem more formally as follows: We assume a universe of elements $\mathcal{U} = \bigcup_{i=1}^{m} \mathcal{U}_i$ partitioned into m non-overlapping groups, a metric distance function d defined for any two pairs of elements, and a set of fairness constraints $\langle k_1, k_2, \cdots, k_m \rangle$, where each k_i is a non-negative integer with $k_i \leq |\mathcal{U}_i|$. Our goal is to select a set $\mathcal{S} \subseteq \mathcal{U}$ of size $k = \sum_{i=1}^{m} k_i$, such that $|\mathcal{S} \cap \mathcal{U}_i| = k_i$ for all i, and such that the minimum distance of any two items in $\mathcal{S}$ is maximized. In this paper, we show that fair Max-Min diversification is NP-complete, and we contribute efficient algorithms with strong approximation guarantees in the case of non-overlapping groups; we further generalize our results and analysis to the case of overlapping groups. We list our contributions at the end of this section.

Contrast with prior work and related problems

Our work augments the existing literature of traditional problems that have been studied under *group fairness* constraints, such as clustering [18, 35], ranking systems [14, 48, 49, 50] and set selection [45]. We proceed to review prior work in closely-related problems and describe how our contributions augment the existing literature. (Summary shown in Figure 2a.)

Max-Min and Max-Sum diversification. The unconstrained version of Max-Min diversification is a special case of our fair variant for $m = 1$. This problem was initially studied in the operation research literature under the name *remote-edge* or *p-dispersion*, along with another popular diversity model, the *Max-Sum* or *remote-clique* model [16, 26, 30, 36, 43]. Similar formulations have also been studied in the context of obnoxious facility location on graphs [46]. While the Max-Min model aims to maximize the minimum pairwise distance in the selected set, the Max-Sum model aims to maximize the total sum of pairwise distances in a set of k items. Max-Sum, as an additive objective, is easier to analyze but tends to select points at the limits of the data space and thus it is not well-suited to applications that require more uniform coverage (see example in Figure 2b). The unconstrained diversification problems are NP-complete even in metric spaces but, for both, a greedy algorithm offers a $\frac{1}{2}-$factor approximation, that has also been shown to be tight [6, 8, 43].

(a) Example 1. **(b)** Example 2.

Figure 3 (a) An example where no optimal solution for the clustering problem is optimal for the diversity problem and vice versa. Suppose we have to pick one white point and one black point. The unique optimal solution for clustering is $\{2, 5\}$ whereas the unique optimal solution for Max-Min diversity is $\{1, 6\}$. (b) An optimal solution for the clustering problem may be arbitrarily bad for the diversity problem. Suppose we have to pick one white and one black point. Set $\{2, 3\}$ is an optimal solution for clustering but yields an arbitrarily bad approximation ratio for the diversity problem as points 2 and 3 can be arbitrarily close together.

Fair Max-Sum diversification. Abbassi et al. [1] study the *fair* Max-Sum diversification problem (assuming disjoint groups) under matroid constraints, where the retrieved subset needs to be an independent set of a matroid of size k (we discuss the correspondence between group fairness constraints and partition matroids in Appendix B). They propose a local search algorithm with a $\left(\frac{1}{2} - \epsilon\right)$-approximation guarantee. Borodin et al. [8, 9] study a bi-criteria optimization problem formulated as the sum of a submodular function and the Max-Sum diversification objective under matroid constraints. They show that the local search approach preserves the $\left(\frac{1}{2} - \epsilon\right)$-approximation guarantee. In an effort to make the state-of-the-art local search algorithms more efficient, Ceccarello et al. [11] propose algorithmic approaches for constructing core-sets with strong approximation guarantees, resulting in efficient algorithms with comparable quality to the best known local search algorithms [1, 8, 9]. A core-set is a small subset of the original data set that contains an α-approximate solution for the Max-Sum diversification problem. Cevallos et al. [15] extend the local search approach to distances of negative type and design algorithms with $O\left(1 - \frac{1}{k}\right)$-approximation and $O(nk^2 \log k)$ running time.

Fair k-center clustering. In the k-center clustering problem the objective is to select k centers such that the maximum distance of any point from its closest cluster center is minimized. Intuitively, cluster centers tend to be distributed in a way that optimizes data coverage. Thus, k-center clustering can serve as another mechanism to perform diverse data selection, albeit the optimization objective is different from Max-Min. Max-Min diversification and k-center clustering are closely related. In fact, the approximation algorithms by Gonzalez [29] for the clustering problem and by Ravi et al. [43] and Tamir [46] for Max-Min diversification, are all based on the same farthest-first traversal heuristic, and they all provide a $\frac{1}{2}$-approximation guarantee. Nonetheless, the analysis of the two algorithms is substantially different and it is not always the case that an algorithm for one problem is applicable to the other.

In recent work, Kleindessner et al. [35] introduced the fair variant of the problem, where the centers are partitioned into m different groups and the constraint of selecting k_i elements per group is enforced in the output of the process. It is easy to find examples where no optimal solution for the fair k-center problem is optimal for the Max-Min objective and vice versa (see example in Figure 3a). Furthermore, we note that an optimal solution for fair k-center clustering can be arbitrarily bad for the Max-Min objective (e.g., Figure 3b). Consequently, the two problems need to be studied independently. The fair k-center clustering problem can also be expressed by a partition matroid, for which Chen et al. [17] provide a 3-approximation algorithm with a quadratic runtime. Kleindessner et al. [35] provide a linear-time algorithm with a $\left(3 \cdot 2^{m-1} - 1\right)$-approximation, while more recent work improved

this bound to $3(1+\epsilon)$ [20], and 3-approximation [33]. In our Appendix, by adapting the ideas for fair Max-Min diversification, we design a linear-time algorithm for fair k-center clustering that also achieves a constant 3-factor approximation.

Outline of contributions: Fair Max-Min diversification. To the best of our knowledge, this paper is the first to introduce fairness constraints to Max-Min diversification. We initially focus on the case of disjoint groups, but extend our algorithms to tackle the overlapping case as well. Our work makes the following contributions.

- After some background and preliminaries, we introduce and formally define the problem of *fair* Max-Min diversification focusing on non-overlapping groups, and further discuss its complexity and approximability (Section 2). To the best of our knowledge, no prior work has studied the Max-Min diversification objective under fairness constraints. In our Appendix, we also describe how our algorithmic frameworks support any constraints that can be expressed in terms of partition matroids (Appendix B).
- We propose a swap-based greedy approximation algorithm, with linear runtime, for the case of $m = 2$, which offers a constant $\frac{1}{4}$-factor approximation guarantee (Section 3.1).
- We propose a general max-flow-based polynomial algorithm, with runtime linear in the size of the data, that offers a $\frac{1}{3m-1}$-factor approximation (Section 3.2). We also demonstrate that for constant m and *small* values for $k = o(\log n)$, we can achieve a constant $\frac{1}{5}$-approximation, also in linear time. While this bound is obviously stronger than our bound for the general case, the $\frac{1}{5}$-approximation algorithm becomes impractical as k increases (Section 3.3).
- We generalize the *fair* diversification problem to the case of overlapping groups (an element can belong to multiple demographic groups). We propose polynomial-time algorithms with $\frac{1}{4}$-factor approximation for the case of $m = 2$ and $\frac{1}{3\binom{m}{\lfloor m/2 \rfloor}-1}$-factor approximation for any m (Section 4).

2 Fair Max-Min Diversification: Background and Problem Definition

In this section, we review necessary background and preliminaries on the Max-Min diversification objective and relevant approximations. Then, we formally define the *fair* Max-Min diversification problem, which generalizes Max-Min diversification. We further characterize the hardness of the problem, and the hardness of its approximation.

2.1 Max-Min Diversification

Problem definition. Prior work has identified a range of diversity objectives to perform diverse data selection. In this work we primarily focus on the Max-Min objective, which corresponds to the minimum distance of any two items in a set $\mathcal{S}$. More formally, we assume a universe of elements $\mathcal{U}$ of size n, a positive integer $k \leq |\mathcal{U}|$ and a pseudometric distance function $d : \mathcal{U} \times \mathcal{U} \to \mathbb{R}_0^+$ that satisfies the following properties for every $u, v \in \mathcal{U}$: $d(u,u) = 0$, $d(u,v) = d(v,u)$ (symmetry), and $d(u,v) \leq d(u,w) + d(w,v)$ (triangle inequality). Then, $d(u,v)$ captures the dissimilarity of the elements $u, v \in \mathcal{U}$, and the Max-Min diversity score of a set $\mathcal{S}$ is $\text{div}(\mathcal{S}) = \min_{u,v \in \mathcal{S}, u \neq v} d(u,v)$. Max-Min diversification seeks to identify a set $\mathcal{S} \subseteq \mathcal{U}$ and $|\mathcal{S}| = k$, such that the minimum pairwise distance, $\text{div}(\mathcal{S})$, of elements in $\mathcal{S}$ is maximized.

Algorithm 1 GMM Algorithm.

Input: $\mathcal{U}$: Universe of available elements
$k \in \mathbb{Z}_0^+$
I: An initial set of elements
Output: $\mathcal{S} \subseteq \mathcal{U}$ of size k

procedure GMM($\mathcal{U}, I, k$)
  $\mathcal{S} \leftarrow \emptyset$.
  if $I = \emptyset$ **then**
    $S \leftarrow$ an arbitrarily chosen point in $\mathcal{U}$
  while $|\mathcal{S}| < k$ **do**
    $x \leftarrow \underset{u \in \mathcal{U}}{\text{argmax}} \ \min_{s \in \mathcal{S} \cup I} d(u, s)$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{x\}$
return $\mathcal{S}$

Algorithms and approximations. This problem formulation was initially studied in the operation research literature by Ravi et al. [43] and in the context of facility location on graphs by Tamir [46]. They both show that the problem is NP-complete even in metric spaces and give a greedy algorithm, GMM, that guarantees a $\frac{1}{2}$-approximation for Max-Min diversification. Ravi et al. [43] also show that this problem cannot be approximated within a factor better than $\frac{1}{2}$ unless P=NP through a reduction from the clique problem.

The GMM approximation algorithm uses the simple and intuitive farthest-first traversal heuristic: Given a set of items $\mathcal{S}$, add the element from $\mathcal{U}$ whose minimum distance from any element in $\mathcal{S}$ is the largest. Algorithm 1 shows the pseudocode for GMM, which starts with an initial set of elements I and greedily augments it with k elements from $\mathcal{U}$. Note that the GMM algorithm, as presented by Ravi et al. [43] and Tamir [46] assumes that $I = \emptyset$; in this paper, we use the slight variant presented in Algorithm 1, which assumes that I can be non-empty. We use GMM as a building block for the algorithms we present in this paper. A naive implementation of the algorithm requires $O((|I| + k)^2 n)$ time but more efficient implementation requires $O((|I| + k)n)$ time; see, e.g., [35, 47] for details.

2.2 Fair Max-Min Diversification

Problem definition and analysis. We assume a universe of elements $\mathcal{U}$ of size n, comprising of m non-overlapping classes: $\mathcal{U} = \bigcup_{i=1}^{m} \mathcal{U}_i$; we further assume a pseudometric distance function $d : \mathcal{U} \times \mathcal{U} \to \mathbb{R}_0^+$; finally, we assume non-negative integers $\langle k_1, \ldots, k_m \rangle$, which we call *fairness constraints*. Our goal is to identify a set $\mathcal{S} \subseteq \mathcal{U}$, such that for all i, $|\mathcal{S} \cap \mathcal{U}_i| = k_i$, and the minimum distance of any two items in $\mathcal{S}$ is maximized. More formally:

$$\textsc{Fair Max-Min}: \underset{\mathcal{S} \subseteq \mathcal{U}}{\text{maximize}} \quad \min_{\substack{u,v \in \mathcal{S} \\ u \neq v}} d(u, v)$$
$$\text{subject to } |\mathcal{S} \cap \mathcal{U}_i| = k_i, \ \forall i \in [m]$$

Intuitively, Fair Max-Min aims to derive the set with the maximum diversity score $\text{div}(\mathcal{S})$, while satisfying the fairness constraints.[2] Next, we state formally the hardness of Fair Max-Min and bound its approximability. These results follow easily from the corresponding prior results on unconstrained Max-Min diversification, as that problem reduces to Fair Max-Min for $m = 1$. We give the proof of Corollary 1 in Appendix A.

[2] A formulation of the fairness constraints with inequalities ($\geq k_i$) would be essentially equivalent: since the diversity score can only decrease as the number of selected points increases, the optimal solution would always select the minimum number of points allowed by the constraints.

▶ **Corollary 1** (Hardness and Approximability Bound). *Determining if there exists a solution to* Fair Max-Min *with diversity score* $\geq \delta$ *is NP-complete. Further, there exists no polynomial-time* α*-approximation algorithm for* Fair Max-Min *with* $\alpha > \frac{1}{2}$*, unless P=NP.*

Our contributions to this problem. To the best our knowledge, this is the first paper to augment the Max-Min diversification problem with fairness constraints. For this problem, typically m is a small constant and $k \ll n$. Therefore, when considering algorithmic complexity, we want to avoid high-order dependence on the size of the data, n. In Section 3, we provide linear-time algorithms, with respect to n, with strong approximation guarantees for this problem in the case of non-overlapping groups. In Section 4, we extend our results to design polynomial-time algorithms with strong approximation guarantees for the generalized setting of overlapping groups.

3 Approximating Diversity

In Section 2.2, we showed that the *fair* formulation for the Max-Min diversification problem is NP-hard, and cannot be approximated within a factor better than $\frac{1}{2}$. In this section, we propose three approximation algorithms for this problem, with a best overall bound of $\frac{1}{4}$ for the case of $m = 2$. For ease of exposition, in the rest of the paper we frequently refer to each of the m groups as different colors.

Our algorithms use GMM (Algorithm 1) as a building block, but adapting GMM for fair Max-Min diversification is not straightforward. We give an example of a simple and intuitive algorithm based on GMM that can lead to an arbitrarily bad result, even in the case of $m = 2$ colors. In the first phase of the algorithm, we use GMM to greedily select elements of any color until the constraints for one of them are satisfied. In the second phase, we allow GMM to greedily select the remaining elements only from the under-satisfied color. Suppose that our data consist of one white and three black elements positioned in a line as follows:

Further, consider that the fairness constraints require the selection of one white and two black elements, and that GMM first selects a black element. Regardless of which black element is selected first, the simple algorithm we described will always be forced to select elements 1 and 2 – the possible selection scenarios are: $\{1, 4, 2\}$, $\{3, 1, 2\}$, and $\{4, 1, 2\}$ – which can be arbitrarily close to one another. This example demonstrates how the choices made for one color, can lead to arbitrarily bad choices for the other color(s), and the problem gets harder as m increases.

Our algorithms employ GMM in ways that guarantee the preservation of *good* choices for all colors. We start with a swap-based algorithm that offers a $\frac{1}{4}$ approximation when $m = 2$. Then we present a flow-based algorithm with a $\frac{1}{3m-1}$ approximation when $m \geq 3$. Both algorithms run in $O(kn)$ time. Finally, we present a $\frac{1}{5}$-approximation for $m \geq 3$ that also runs in $O(kn)$, on the assumption m is constant and $k = o(\log n)$. However, the running time of this third algorithm has an additional factor that depends exponentially on k, which makes the algorithm practical only for *small* k values, e.g., for $n = 10^4$, $k \approx 10$.

3.1 Fair-and-Diverse Selection: $m = 2$

In the binary setting, the input is a set of points $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$ and two non-negative integers $\langle k_1, k_2 \rangle$ with $k_i \leq |\mathcal{U}_i|$ for all $i \in \{1, 2\}$. We want to select a set $\mathcal{S}$ with k_i elements from each $\mathcal{U}_i$ partition such that the div$(\mathcal{S})$ is maximized.

Algorithm 2 FAIR-SWAP: Fair Diversification for $m = 2$.

Input: $\mathcal{U}_1, \mathcal{U}_2$: Set of points of color 1 and 2
$k_1, k_2 \in \mathbb{Z}_0^+$
Output: k_i points in $\mathcal{U}_i$ for $i \in \{1, 2\}$
1: **procedure** FAIR-SWAP
▷Color-Blind Phase:
2: $\mathcal{S} \leftarrow \text{GMM}(\mathcal{U}, \emptyset, k_1 + k_2)$
3: $\mathcal{S}_i = \mathcal{S} \cap \mathcal{U}_i$ for $i \in \{1, 2\}$
▷Balancing Phase:
4: Set $U = \text{argmin}_i(|\mathcal{S}_i| - k_i)$ ▷Under-satisfied set
5: $O = 3 - U$ ▷Over-satisfied set
6: Compute the sets:
$E \leftarrow \text{GMM}(\mathcal{U}_U, \mathcal{S}_U, k_U - |\mathcal{S}_U|)$
$R \leftarrow \{\text{argmin}_{x \in \mathcal{S}_O} d(x, e) : e \in E\}$
return $(\mathcal{S}_U \cup E) \cup (\mathcal{S}_O \setminus R)$

Algorithm and intuition. FAIR-SWAP (Algorithm 2) has two phases; the *color-blind* and the *balancing* phase. In the *color-blind* phase, we call GMM by initializing I to the empty set so as to retrieve a set $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$ of size k (line 2). If $|\mathcal{S}_1| = k_1$ and $|\mathcal{S}_2| = k_2$ then these two sets are returned. Alternatively, if one set is smaller than required, then the other set is larger than required, and we need to rebalance these sets. Let $\mathcal{S}_U$ be the set that is too small and let $\mathcal{S}_O$ be the set that is too large. The algorithm next finds $k_U - |\mathcal{S}_U|$ extra points $E \subseteq \mathcal{U}_U$ to add to $\mathcal{S}_U$ by again using the GMM algorithm, this time initialized with the set $\mathcal{S}_U$. For each point in E we then remove the closest point in $\mathcal{S}_O$ (line 6). In this way we add $k_U - |\mathcal{S}_U|$ points to $\mathcal{S}_U$ and remove $k_U - |\mathcal{S}_U|$ points from $\mathcal{S}_O$. After this rebalancing the size of $\mathcal{S}_U$ is $|\mathcal{S}_U| + (k_U - |\mathcal{S}_U|) = k_U$ and the size of $\mathcal{S}_O$ is $|\mathcal{S}_O| - (k_U - |\mathcal{S}_U|) = k - k_U = k_O$ as required. Note that sets E and R will be empty if the sets are already balanced after the color blind phase and thus the set $\mathcal{S}$ will not be altered by the balancing phase.

Running-time analysis. The running time of FAIR-SWAP (Algorithm 2) is $O(kn)$. In the color-blind phase of the algorithm we run GMM on $\mathcal{U}$ with $I = \emptyset$ and this takes $O(kn)$ time. Then in the balancing phase, computing the extra points E via the GMM algorithm takes $O(kn)$ time and computing R takes $O(k^2)$ time since there are fewer than k points in E and at most k points in $\mathcal{S}_O$.

Approximation-factor analysis. Let $\mathcal{S}^*$ be the set of k points in $\mathcal{U}$ that maximize the diversity when there are no fairness constraints. Let $\ell^* = \text{div}(\mathcal{S}^*)$. Let $\mathcal{F}^* = \mathcal{F}_1^* \cup \mathcal{F}_2^*$ be the set of k points in $\mathcal{U}$ that maximize the diversity subject to the constraint that for each $i \in \{1, 2\}$, k_i points are chosen of color i. Let $\ell^*_{\text{fair}} = \text{div}(\mathcal{F}^*)$ and note that $\ell^* \geq \ell^*_{\text{fair}}$.

We first argue that $\text{div}(\mathcal{S}) \geq \ell^*/2 \geq \ell^*_{\text{fair}}/2$. This follows because, by the triangle inequality, there is at most one point in $\mathcal{S}^*$ that is distance $< \ell^*/2$ from each point in $\mathcal{S}$; otherwise two points in $\mathcal{S}^*$ would be $< \ell^*$ apart and this contradicts the fact $\text{div}(\mathcal{S}^*) = \ell^*$. Hence, while the GMM algorithm has picked $< k$ elements, there exists at least one element in $\mathcal{S}^*$ that can be selected that is distance $\geq \ell^*/2$ from all the points already selected. Since the algorithm picks the next point farthest away from the points already chosen, the next point is at least $\ell^*/2$ from the existing points.

Next we argue that $\text{div}(\mathcal{S}_U \cup E) \geq \ell^*_{\text{fair}}/2$. To show this, first observe that, $\text{div}(\mathcal{S}_U) \geq \text{div}(\mathcal{S}) \geq \ell^*_{\text{fair}}/2$. Next consider the points added to E by GMM. By the triangle inequality there is at most one point in $\mathcal{F}_U^*$ that is distance $< \ell^*_{\text{fair}}/2$ from each point in $\mathcal{S}_U \cup E$. Hence,

while GMM has picked $< k_U - |\mathcal{S}_U|$ elements, there exists at least one element that can be selected that is distance $\geq \ell^*_{\text{fair}}/2$ from the points already selected. Since the algorithm picks the next point farthest away from the points already chosen, the next point is at least $\ell^*_{\text{fair}}/2$ from the existing points. Thus, we can guarantee that $d(x, y) \geq \ell^*_{\text{fair}}/2$ for all pairs of points $x, y \in \mathcal{S}_U \cup E \cup \mathcal{S}_O$ except potentially when $x \in E$ and $y \in \mathcal{S}_O$.

To handle this case, for each $x \in E$ we remove the closest point in $\mathcal{S}_O$. Note that by an application of the triangle inequality and the fact that $\text{div}(\mathcal{S}_O) \geq \ell^*_{\text{fair}}/2$, for each $x \in E$ there can be at most one point $y \in \mathcal{S}_O$ such that $d(x, y) < \ell^*_{\text{fair}}/4$. Hence, after the removal of the closest points the distance between all pairs is $\geq \ell^*_{\text{fair}}/4$ as required. We summarize the analysis of this section as follows:

▶ **Theorem 2.** *Fair-Swap (Algorithm 2) is a $1/4$-approximation algorithm for the fair diversification problem when $m = 2$ that runs in time $O(kn)$.*

Connections to prior art. The idea of balancing has also been successfully applied to matroid optimization settings subject to fairness constraints [19], and to the red-blue matching problem [39]. However, our objective function cannot be expressed by a matroid (or an intersection of matroids), and thus the approaches of prior work are not applicable to our setting. Further, the algorithms and analysis are distinct for these problems; Fair-Swap builds upon GMM while the algorithms designed in [19] employ the Edmonds algorithm for finding a maximum independent set.

3.2 Fair-and-Diverse Selection: $m \geq 3$

Basic algorithm. We start by presenting a basic algorithm that takes as input a guess γ for the optimum fair diversity. If this guess is greater than the optimum fair diversity then the algorithm may abort, but if the algorithm does not abort, it will return a fair diversity at least $\gamma/(3m - 1)$.

Algorithm and intuition. The approach of Fair-Flow (Algorithm 3) is to construct disjoint sets of points $C_1, C_2, \ldots$ such that, if γ is at most the optimal fair diversity, it is possible to find sets $\mathcal{S}_1, \ldots, \mathcal{S}_m$ of sizes $k_1, \ldots, k_m$ such that each C_i contains at most one point from $\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_m$. If we can construct $C_1, C_2, \ldots$ such that for any $x \in C_i$ and $y \in C_j$, then $d(x, y) \geq d_2$ for some value d_2 then we have $\text{div}(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_m) \geq d_2$. Furthermore, because the sets $C_1, C_2, \ldots$ are disjoint it is possible to find sets $\mathcal{S}_1, \ldots, \mathcal{S}_m$ with the required property via a reduction to network flow (noting that the optimal flow in a network with integer capacities is always integral). See the algorithm for the precise reduction and see Figure 4 for an example.

The way we construct each $C_1, C_2, \ldots$ is to first run GMM on each color class i and use this to identify at most k points Z_i of color i such that $\text{div}(Z_i) \geq d_1$ for some value d_1 to be determined. We then partition $\bigcup_i Z_i$ into the disjoint groups $C_1, C_2, \ldots$ where the partition satisfies the property that any two points $x, y \in \bigcup_i Z_i$ such that $d(x, y) < d_2$ are in the same group. Note that x, z will end up in the same group if there exists y such that $d(x, y) < d_2$ and $d(y, z) < d_2$; more generally two points can end up in the same group because of a chain of points where each adjacent pair of points are close. However, in the analysis, we will show that these chains cannot be too long and, for appropriately chosen d_1 and d_2, any two points in C_j are distance $< d_1$ from each other. In the analysis, this will enable us to argue that if γ is at most the optimal fair diversity, it is possible to find the required sets $\mathcal{S}_1, \ldots, \mathcal{S}_m$.

Algorithm 3 FAIR-FLOW: Fair Diversification for $m \geq 3$.

Input: $\mathcal{U}_1, \ldots, \mathcal{U}_m$: Universe of available elements
$k_1, \ldots, k_m \in \mathbb{Z}_0^+$
$\gamma \in \mathbb{R}$: A guess of the optimum fair diversity
Output: k_i points in $\mathcal{U}_i$ for $i \in [m]$

procedure FAIR-FLOW
  for $i \in [m]$ **do**
    $Y_i \leftarrow \text{GMM}(\mathcal{U}_i, \emptyset, \sum_i k_i)$
  $Z_i \leftarrow$ maximal prefix of Y_i such that all points in Z_i are $\geq d_1 = \frac{m\gamma}{3m-1}$ apart.
  Construct undirected graph G_Z with nodes $Z = \bigcup_i Z_i$ and edges (z_1, z_2), if $d(z_1, z_2) < d_2 = \frac{\gamma}{3m-1}$.
  $C_1, C_2, \ldots C_t \leftarrow$ Connected components of G_Z.
  ▷Construct flow graph
  Construct directed graph $G = (V, E)$ where

$$\begin{aligned} V &= \{a, u_1, \ldots, u_m, v_1, \ldots, v_t, b\} \\ E &= \{(a, u_i) \text{ with capacity } k_i : i \in [m]\} \\ &\quad \cup \{(v_j, b) \text{ with capacity } 1 : j \in [t]\} \\ &\quad \cup \{(u_i, v_j) \text{ with capacity } 1 : |Z_i \cap C_j| \geq 1\} \end{aligned}$$

  Compute max a-b flow.
  if flow size $< k = \sum_i k_i$ **then return** $\emptyset$ ▷Abort
  else ▷max flow is k
    $\forall (u_i, v_j)$ with flow add a node in C_j with color i to $\mathcal{S}$.
  return $\mathcal{S}$

Analysis of basic algorithm. We need a preliminary lemma that argues that all the points in the same connected component are close together.

▶ **Lemma 3.** *For all connected components C_j, $\forall x, y \in C_j : \ d(x, y) < (m-1)d_2$ and C_j does not contain any two points of the same color.*

Proof. Consider two points $x, y \in C_j$ and let the length of a shortest unweighted path $P_{x,y}$ between x and y in the graph be ℓ. If $\ell \leq m-1$ then $d(x, y) < (m-1)d_2$ as required. If $\ell \geq m$ then there exists two points on this path (including end points) that have the same color and this will lead to a contradiction. Consider the subpath $P_{x',y'} \subset P_{x,y}$ where x' and y' have the same color i and all internal nodes have distinct colors. Then the length of $P_{x',y'}$ is strictly less than $md_2 = m\gamma/(3m-1) = d_1$. But this contradicts $d(x', y') \geq d_1$ for all points in Z_i. ◀

The next theorem establishes that when the algorithm does not abort, the solution returned has diversity at least $\gamma/(3m-1)$ and that it never aborts if the guess γ is at most the optimum diversity.

▶ **Theorem 4.** *Let ℓ^*_{fair} be the optimum diversity. If $\gamma \leq \ell^*_{fair}$ then the algorithm returns a set of points of the required colors that are each $\geq \gamma/(3m-1)$ apart. If $\gamma > \ell^*_{fair}$ then the algorithm either aborts or returns a set of points of the required colors that are each $\geq d_2 = \gamma/(3m-1)$ apart.*

Proof. Note that if the algorithm does not abort then all points are $\geq \gamma/(3m-1)$ apart since any two points in different connected components are $\geq \gamma/(3m-1)$ apart.

Hence, it remains to argue that if $\gamma \leq \ell^*_{\text{fair}}$ then the algorithm does not abort. To argue this, we will construct a flow of size k in the network instance. And to do this it suffices to identify k_i connected components including a point from Z_i for each i, such that the resulting set of $k_1 + k_2 + \ldots + k_m$ connected components are all distinct. To do this, we start by defining a node u_i to be critical if $|Z_i| < k$ and non-critical otherwise. Let $O_i \subset \mathcal{U}_i$ be the set of k_i points in the optimum solution. For $x \in \mathcal{U}_i$, let $f(x)$ be the closest point Z_i to x. If

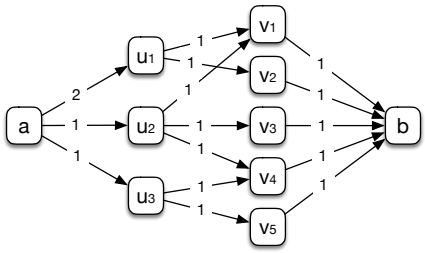


Figure 4 The graph construction in Algorithm 3 (line 7) corresponding to $m = 3$, $k_1 = 2, k_2 = 1, k_3 = 1$. Points of color 1 are contained in C_1 and C_2. Points of color 2 are contained in C_1, C_3, and C_4. Points of color 3 are contained in C_4 and C_5. Note there is an a-b flow of size $k_1 + k_2 + k_3$ iff it is possible to pick at most one point from each C_j while still picking at most k_i points of color i for each $i \in [m]$.

u_i is critical, then note that $d(x, f(x)) < d_1$. Note that for all points $x, y \in \cup_{i:\text{critical}} f(O_i)$, $d(x, y) > \ell^*_{\text{fair}} - 2d_1 \geq \gamma - 2\gamma m/(3m-1) = (m-1)d_2$ and hence, by Lemma 3, this implies that all points in $\bigcup_{i:\text{critical}} f(O_i)$ are in different connected components.

We then consider each non-critical node u_i in turn. Since u_i was non-critical and each connected component has at most one point in each Z_i, there are k connected components that include a point in Z_i. At most $k - k_i$ need to be used to pick points of other classes and hence at least $k - (k - k_i) = k_i$ remain. ◀

Final algorithm. Our final algorithm is based on binary searching for a "good" guess γ for the optimum diversity ℓ^*_{fair} where each guess can be evaluated using the basic algorithm above. The goal is to find a guess that is close to ℓ^*_{fair} or larger such that the algorithm does not abort. There are two natural ways to do this; which is best depends on parameters of the data set.

Binary-searching over continuous range: For the first approach, note that $\ell^*_{\text{fair}} \in [d_{min}, d_{max}]$ where $d_{\min} = \min_{x,y \in \mathcal{U}: x \neq y} d(x, y)$, and $d_{\max} = \max_{x,y \in \mathcal{U}: x \neq y} d(x, y)$. Hence, there exists a guess $\gamma = (1+\epsilon)^i d_{\min}$ for some $i \in \{0, 1, 2, \ldots, \lceil \log_{1+\epsilon} R \rceil\}$ where $R := d_{\max}/d_{\min}$ such that $\ell^*_{\text{fair}}/(1+\epsilon) \leq \gamma \leq \ell^*_{\text{fair}}$. Note that for this guess, the algorithm returns a $(3m-1)(1+\epsilon)$ approximation. We can find this guess (or an even better guess, i.e., a $\gamma > \ell^*_{\text{fair}}$ for which the algorithm does not abort) via a binary search over the $1 + \lceil \log_{1+\epsilon} R \rceil$ possible guesses. The number of trials required is $O(\log(1 + \lceil \log_{1+\epsilon} R \rceil) = O(\log(\epsilon^{-1}) + \log\log R)$.

Binary-searching over discrete set: For the second approach we note that after the algorithm's initial step (which did not depend on the guess γ) there are only km points and hence at most $\binom{km}{2}$ distinct distances between remaining points. Hence, it suffices to only consider guesses γ such that d_1 or d_2 corresponds to one of these $O(k^2m^2)$ values. We can sort these values in $O(k^2m^2 \log km)$ time and then binary search over this range to find a good guess using $O(\log km)$ trials.

Final diversification result. Our main theorem of this section follows by combining the binary search over a discrete set approach with the basic algorithm.

▶ **Theorem 5.** *There is a $\frac{1}{3m-1}$-approximation algorithm for the fair diversity problem that runs in time $O(kn + k^2m^2 \log(km))$.*

Proof. The time to construct $Y_1, \ldots, Y_m$ is $O(kn)$. We then need to sort the $O(k^2m^2)$ distances amongst these points. This takes $O(k^2m^2 \log(km))$ time. The time to construct and solve the flow instance is $O(k^2m^2)$ since the flow instance has $O(km)$ nodes and $O(km)$ edges [40, 41]. Note that the binary search requires us to construct and solve $O(\log(km))$ flow instances. Hence the total running time is as claimed. ◀

Algorithm 4 FAIR-GMM: Fair Diversification for small k.

Input: $\mathcal{U}_1, \ldots, \mathcal{U}_m$: Universe of available elements
$k_1, \ldots, k_m \in \mathbb{Z}_0^+$
Output: k_i points in $\mathcal{U}_i$ for $i \in [m]$
1: **procedure** FAIR-GMM
2: **for** $i \in [m]$ **do** $Y_i \leftarrow \text{GMM}(\mathcal{U}_i, \emptyset, \sum_i k_i)$
3: By exhaustive search, find the sets $\mathcal{S}_i \subseteq Y_i$ for $i \in [m]$ such that $|\mathcal{S}_i| = k_i$ and $\text{div}(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_m)$ is maximized.

If we used the binary search over a continuous range approach, the running time would by $O(kn + k^2m^2(\log \epsilon^{-1} + \log\log d_{\max}/d_{\min}))$ and the approximation ratio would be $\frac{1}{(3m-1)(1+\epsilon)}$.

3.3 Fair-and-Diverse Selection: Small k, m

In this section, we present a simple algorithm that has the advantage of achieving a better approximation ratio than the algorithm in the previous section. The downside of the algorithm is that the running time is exponential in k, specifically, $O(kn + k^2(em)^k)$. However, when $m = O(1)$ and $k = o(\log n)$ the dominating term in the running time is $O(kn)$, as in the case of the algorithms from the previous sections.

Algorithm and intuition. The basic approach of FAIR-GMM (Algorithm 4) is to first select k points (or less if there are fewer than k points of a particular color) of each color via the GMM algorithm. The resulting subset $\bigcup_i Y_i$ has at most km points and this is significantly smaller than the original set of points assuming k and m are much smaller than n. Hence, it is feasible to solve the problem via exhaustive search on the subset of points. In the analysis, we will be able to show that the optimal fair diversity amongst the subset of points is at least $1/5$ of the optimal fair diversity amongst $\bigcup_i \mathcal{U}_i$.

Analysis. To prove the approximate factor we need to show that the optimal solution amongst the subset of points selected in step one has diversity that is not significantly smaller than the optimal diversity of the original set of points. To show this the basic idea is that for each i, the set Y_i will contain at least one point near every color i point in the optimal solution or will contain k points such that even if we remove any set of $k - k_i$ points to make space for points of other colors, the remaining set of k_i points of color i still has sufficiently high diversity.

▶ **Theorem 6.** *Algorithm 4 returns a $\frac{1}{5}$-approximation and the running time is $O(kn + k^2(em)^k)$. Note that this is $O(kn)$ when $k = o(\log n)$ and $m = O(1)$.*

Proof. For the running time, note that Step 1 can be implemented in $O(kn)$ time. For Step 2, note that there are at most km points in $Y_1, Y_2, \ldots Y_m$ so a brute force algorithm needs to consider at most $\binom{km}{k} \leq (em)^k$ sets of points and computing the min distance for each takes $O(k^2)$ time. Note that this is $o(n)$ assuming $k = o(\log n)$ and m is constant.

For the approximation ratio, it suffices to argue that if ℓ^*_{fair} is the optimum value then there exists a set of points amongst $Y_1 \cup \ldots \cup Y_m$ with the required colors that are $\ell^*_{\text{fair}}/5$ apart. Let Z_i be the maximal prefix of Y_i such that all points at points are $\geq 2\ell^*_{\text{fair}}/5$ apart. For each $x \in \mathcal{U}_i$, let $f(x)$ be the closest point in Z_i. Call i critical if $|Z_i| < k$. Note that if i is critical, then $d(x, f(x)) < 2\ell^*_{\text{fair}}/5$. Let O_i be the optimal set of color i points and consider the subsets $\mathcal{S}_1, \mathcal{S}_2, \ldots \mathcal{S}_m$ of points in $Z_1, Z_2, \ldots Z_m$ defined as follows:

- For all i that are critical, let $\mathcal{S}_i = f(O_i)$ and let $D = \cup_{i:\text{critical}} \mathcal{S}_i$. Note that $\text{div}(D) > \ell^*_{\text{fair}} - 4\ell^*_{\text{fair}}/5 = \ell^*_{\text{fair}}/5$.

Figure 5 An example with $m = 2$ overlapping classes, with $|\mathcal{U}_1| = 3$ and $|\mathcal{U}_2| = 4$, where (a) the fairness constraints can be satisfied with fewer than k elements and (b) a class has to be overrepresented to satisfy the fairness constraints for all classes. Suppose we have to pick two white and one black element ($k = 3$). A feasible solution consists of two bi-colored elements, thus fewer than k, in which the black class is represented by two and not just one element.

- For each j that is not critical: Remove all points in Z_j that are distance $< \ell^*_{\text{fair}}/5$ from a point in D. Note that at most one point in Z_j is $< \ell^*_{\text{fair}}/5$ from each point in D because points in Z_j are $\geq 2\ell^*_{\text{fair}}/5$ apart. Hence, at most $|D|$ points are removed from Z_j.
- Process the non-critical j in arbitrary order: Pick k_j points $\mathcal{S}_j$ arbitrarily from Z_j. Remove all points from Z that are distance $< \ell^*_{\text{fair}}/5$ from a point in $\mathcal{S}_j$. This removes at most k_j points from each Z_i. Note that when we process j there are at least $k - (\sum_{i:\mathcal{S}_i \text{ defined so far}} k_i) \geq k - (k - k_j) = k_j$ points in Z_j.

Note $\text{div}(\bigcup_i \mathcal{S}_i) \geq \ell^*_{\text{fair}}/5$ and this implies the claimed approximation factor. ◀

4 Generalizing to Overlapping Groups

In this section, we show how we can extend our algorithmic framework to allow the elements in the universe $\mathcal{U}$ to belong to multiple classes, e.g., an individual may belong to multiple demographic groups such as multiple races, or combinations of race, gender, and other sensitive demographics. First, we formally define the problem and show how our Fair-Swap and Fair-Flow algorithms can be adapted to support this generalized setting.

We assume a universe of elements $\mathcal{U}$ comprising of m *possibly* overlapping classes $\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_m$, a pseudometric distance function $d : \mathcal{U} \times \mathcal{U} \to \mathbb{R}^+_0$ and a set of fairness constraints $\langle k_1, \ldots, k_m \rangle$ where each k_i is a non-negative integer with $k_i \leq |\mathcal{U}_i|$. Our goal is to identify a set $\mathcal{S} \subseteq \mathcal{U}$ to satisfy the fairness constraints such that the minimum distance of any two items in $\mathcal{S}$ is maximized.

It will be convenient to introduce some additional notation. For any $L \subset [m]$, define $X_L = \left(\bigcap_{i \in L} \mathcal{U}_i\right) \cap \left(\bigcup_{j \notin L} \mathcal{U}_j\right)$. That is, X_L consists of all elements exactly in the classes of L and no others. Note that if we select an element in X_L it contributes to helping satisfy $|L|$ of the fairness constraints. Hence, it may be possible to satisfy all the constraints by picking fewer than $k_1 + \ldots + k_m$ elements. Further, a feasible solution may require more than k_i elements for class i (example in Figure 5). Formally, we define the problem as follows:

$$\text{Fair}^+ \text{ Max-Min}: \underset{\mathcal{S} \subseteq \mathcal{U}}{\text{maximize}} \quad \min_{\substack{u,v \in \mathcal{S} \\ u \neq v}} d(u, v)$$
$$\text{subject to } |\mathcal{S} \cap \mathcal{U}_i| \geq k_i, \ \forall i \in [m]$$

4.1 Fair-and-Diverse Selection (Overlaps): $m = 2$

In the binary setting, the input is a set of points $\mathcal{U}$ that comprises of $m = 2$ overlapping classes; $\mathcal{U}_1 = X_{\{1\}} \cup X_{\{1,2\}}$ and $\mathcal{U}_2 = X_{\{2\}} \cup X_{\{1,2\}}$. We design a swap-based algorithm, with 1/4-approximation guarantee, which uses the idea of binary searching over a discrete set of guesses for the optimum fair diversity, denoted as ℓ^*_{fair}.

Algorithm and intuition. The Fair$^+$-Swap algorithm (Algorithm 5) takes as input a guess γ for the optimum fair diversity. We show that if $\gamma \leq \ell^*_{\text{fair}}$, we can always find enough points to construct a fair set $\mathcal{S} = \mathcal{S}_{\{1\}} \cup \mathcal{S}_{\{2\}} \cup \mathcal{S}_{\{1,2\}}$ with $\text{div}(\mathcal{S}) \geq \gamma/4$ (where $\mathcal{S}_L = \mathcal{S} \cap X_L$).

The algorithm first finds as many points as possible in $X_{\{1,2\}}$ and are at least $\frac{\gamma}{4}$ apart from each other. Let $\mathcal{S}_{\{1,2\}}$ be the resulting set, with a total of t points. Note that to satisfy the fairness constraints, we need to add $k_i - t$ points for each class i in $\{1,2\}$. The algorithm proceeds to remove all points in $\mathcal{U}$ that are closer than $\frac{\gamma}{4}$ from any point in $\mathcal{S}_{\{1,2\}}$. It is easy to see that all remaining points, $\mathcal{S}^+$, can only belong to one class, i.e., $\mathcal{S}^+ \cap X_{\{1,2\}} = \emptyset$ (because all points that did not make it to $\mathcal{S}_{\{1,2\}}$ have to be closer than $\frac{\gamma}{4}$ from some point in $\mathcal{S}_{\{1,2\}}$). Since $\mathcal{S}^+$ does not have overlapping classes, we can execute Fair-Swap (Algorithm 2) on it to select a set with $k_i - t$ points for each class i in $\{1,2\}$. In our analysis, we show that $\mathcal{S}^+$ contains at least $k_1 - t$ and $k_2 - t$ points from $X_{\{1\}}$ and $X_{\{2\}}$ that are $\geq \gamma$ apart from each other. Thus, the Fair$^+$-Swap algorithm will produce a set of points that are at least $\gamma/4$ apart from each other.

▶ **Theorem 7.** *Fair$^+$-Swap (Algorithm 5) is a polynomial-time algorithm with* $1/4$*-approximation guarantee for the fair diversification problem with* $m = 2$ *overlapping classes.*

We provide the pseudocode for the Fair$^+$-Swap algorithm (described above) and the proof for Theorem 7 in Appendix A.

4.2 Fair-and-Diverse Selection (Overlaps): $m \geq 3$

The algorithm in this section is an extension of Fair-Flow (Algorithm 3); the previous algorithm did not apply in the case when classes could overlap whereas the new algorithm will. Throughout this section, it will be convenient to use the following notation: $M := \binom{m}{\lfloor m/2 \rfloor}$. The approximation factor for the algorithm designed in this section will be $3M - 1$ in contrast to the $3m - 1$ approximation for the non-overlapping case. Note that for $m = 2, 3, 4, 5$ we have $M = 2, 3, 6, 10$, i.e., when the number of classes is small, M is still relatively small.

There are two main steps that need to be changed in the overlapping case: 1) defining a subset Z of the elements that will be considered and 2) determining how many points to use that appear in multiple classes. We discuss each in turn.

Defining Z. Recall that the first main part of Fair-Flow (Algorithm 3) was to select a subset of points of each color such that all points in each subset was a certain distance apart. When there are overlapping classes, we need to revisit how this is done. Motivated by the fact that an element in $X_{L'}$ contributes to at least as many fairness constraints as an element in X_L if $L \subset L'$, when we select a subset of points in $\mathcal{U}_i$ we want to prioritize points that are also in other classes. For example, for $m = 3$ we have: (1) $\mathcal{U}_1 = X_{\{1\}} \cup X_{\{1,2\}} \cup X_{\{1,3\}} \cup X_{\{1,2,3\}}$, (2) $\mathcal{U}_2 = X_{\{2\}} \cup X_{\{1,2\}} \cup X_{\{2,3\}} \cup X_{\{1,2,3\}}$, and (3) $\mathcal{U}_3 = X_{\{3\}} \cup X_{\{1,3\}} \cup X_{\{2,3\}} \cup X_{\{1,2,3\}}$.

Consistent with "prioritizing points" in multiple classes, we construct subsets of $\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3$ by first constructing a maximal subset $Z_{\{1,2,3\}} \subset X_{\{1,2,3\}}$ such that the pairwise distance of all points is at least d_1. We then define a maximal subset $Z_{\{1,3\}} \subset X_{\{1,3\}}$ such that every point is at least d_1 from each other point in $Z_{\{1,3\}}$ and from points in $Z_{\{1,2,3\}}$. We construct $Z_{\{1,2\}}$ and $Z_{\{2,3\}}$ similarly. Finally $Z_{\{1\}}$ is a maximal subset of $X_{\{1\}}$ such that every point is at least d_1 from each other point in $Z_{\{1\}}$ and from every point in $Z_{\{1,2\}} \cup Z_{\{1,3\}} \cup Z_{\{1,2,3\}}$. Lines 3–5 in Algorithm 6 (given in the Appendix) generalize this process to arbitrary m.

Note that we ensure the property that all points in Z_L are at least d_1 far from each other and from any point in $\bigcup_{L':L\subset L'} Z_{L'}$ but the subset of elements picked from $\mathcal{U}_1$, i.e., $Z_{\{1\}} \cup Z_{\{1,2\}} \cup Z_{\{1,3\}} \cup Z_{\{1,2,3\}} \subset \mathcal{U}_1$, no longer satisfies the condition that they are all at least d_1 far from one another. In particular, there may exist points $x \in Z_L$ and $y \in Z_{L'}$ such that

$d(x,y) < d_1$ if neither L or L' is a subset of the other.[3] A natural question, and an issue that will arise in our analysis is how many sets can there be such that no set is a subset of another. Fortunately, the following classic result in extremal combinatorics resolves this question.

▶ **Lemma 8** (Sperner's Lemma). *A collection of sets is called an* anti-chain *if none of the sets is a subset of another set. If all sets are subsets of $[m]$ then the maximum size of such a collection is $M = \binom{m}{\lfloor m/2 \rfloor}$.*

Next, recall that Fair-Flow (Algorithm 3) then constructs a graph G_Z where the nodes are the selected points and there are edges between points if their distance is $< d_2$. The new algorithm proceeds similarly but with new parameters: $d_1 \leftarrow \frac{M\gamma}{3M-1}$, and $d_2 \leftarrow \frac{\gamma}{3M-1}$. With this setting of the parameters and appealing to Lemma 8 we prove an upper bound on the distance between any two points in the same connected components (proof in Appendix A):

▶ **Lemma 9.** *For all connected components C_j, $\forall x, y \in C_j : \; d(x,y) < (M-1)\, d_2$, and C_j does not contain any two points a, b such that $a \in X_L$ and $b \in X_{L'}$ where $L \subset L'$.*

Guessing how much to exploit points in multiple classes. So far we have (1) discussed how to select the subset Z of input points and (2) partitioned Z such that we have some upper bound on the distance between any two points in the same partition. In the non-overlapping case, we could then argue it suffices to pick at most one point in each partition and adding this point to the output set $\mathcal{S}$ would increment $|\mathcal{S} \cap \mathcal{U}_i|$ for exactly one value $i \in [m]$. In the overlapping case, however, we may need to pick a point in a partition that is in multiple classes and would increment $|\mathcal{S} \cap \mathcal{U}_i|$ for multiple values of i.

To get the reduction to network flow to generalize to the non-overlapping case we need to guess values c_L for every non-empty set $L \subset [m]$ and require that we find at least c_L points in $\cap_{i \in L} \mathcal{U}_i$ such that the $\sum_{L \subseteq [m]} c_L$ points returned are distinct. The fact the points need to be distinct allows the reduction to go through. Note that to satisfy the fairness requirements we need that $\sum_{L: i \in L} c_L \geq k_i$ for each i.

▶ **Example 10.** Suppose we require $k_1 = 2$ points from $\mathcal{U}_1$ and $k_2 = 2$ points from $\mathcal{U}_2$. Then the guess $c_{\{1\}} = 2$ and $c_{\{2\}} = 2$ would correspond to picking at least four distinct points, at least two from $\mathcal{U}_1$ and at least two from $\mathcal{U}_2$. In contrast, the guess $c_{\{1\}} = c_{\{2\}} = 1$, and $c_{\{1,2\}} = 1$ would correspond to picking at least three distinct points where at least one comes from each of sets $\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_1 \cap \mathcal{U}_2$ respectively.

There are at most $k^{2^m - 1 - m}$ possible guesses[4] to try for the values and at least one is feasible since the optimal solution corresponds to some set of guesses. With a feasible set of guesses, we then essentially treat all sets $L \subseteq [m]$ as colors although when we need to pick c_L points of color L, it will suffice to pick points with color L' if L is a subset of L'.

The next theorem establishes that when the algorithm does not abort, the solution returned has diversity at least $\gamma/(3M-1)$ and that it never aborts if the guess γ is at most the optimum diversity.

[3] This is a generalization of the case when there was no-overlap. In that case there could exist $x \in Z_i$ and $y \in Z_j$ such that $d(x,y) < d_1$.

[4] Recall that we typically consider m to be a small constant. A bound of $k^{2^m - 1}$ is immediate because there at most $2^m - 1$ quantities. A slightly tighter bound follows by noting that c_L for all singleton sets L is implied once the other values are chosen.

▶ **Theorem 11.** *Let ℓ^*_{fair} be the optimum diversity. If $\gamma \leq \ell^*_{fair}$ then the algorithm returns a set of points of the required colors that are each $\geq \gamma/(3M-1)$ apart. If $\gamma > \ell^*_{fair}$ then the algorithm either aborts or returns a set of points of the required colors that are each $\geq d_2 = \gamma/(3M-1)$ apart.*

We provide the proof of Theorem 11 in Appendix A. The rest of the algorithm and analysis follows similarly as Algorithm 3, where we binary search for γ in either a continuous or discrete space. The running time is increased by a factor of k^{2^m-m-1} because of the need to guess the values $\{c_L\}_{L\subset[m]}$; thus Fair$^+$-Flow is a polynomial-time algorithm with a $\frac{1}{3\binom{m}{\lfloor m/2\rfloor}-1}$-approximation guarantee.

5 Related Work

Diversity is an important principle in data selection and summarization, facility location, recommendation systems and web search. The diversity models that have been proposed in the literature can be organized into three main categories, (1) the distance-based models where the goal is to minimize the *similarity* of the elements within a set, (2) the coverage-based models where there exists a predetermined number of categories and the aim is to maximize the *coverage* of these categories [4, 38] and (3) the novelty-based models that are defined so as to minimize the *redundancy* of the elements shown to the user [10]. For further information, we refer the reader to the related surveys [23, 24].

Max-Min and Max-Sum diversification are two of the most well studied distance-based models [16, 28, 30, 43], and there exist efficient algorithms with strong approximation guarantees for the unconstrained version of the problems in the offline setting (discussed in Sections 1 and 2). The problem of diversity maximization has also been studied in the streaming and distributed settings, where (composable) core-sets were shown to be a useful theoretical tool [3, 12, 32], and more recently in the sliding window setting [7]. A separate line of work focuses on designing efficient indexing schemes for result diversification [2, 22, 47]; this direction is orthogonal to our work, and it is not clear how to extend existing indexing schemes for fair Max-Min diversification.

There is relatively little prior work on constrained diversification. The closest to our work is fair Max-Sum diversification (discussed in Section 1) and fair k-center clustering (discussed in Section 1 and Appendix C). To the best of our knowledge, our work is the first to augment the traditional Max-Min objective with fairness constraints.

Prior work has also combined fairness with the determinant measure of diversity [13]. That work models fairness constraints the same way as we do, but their algorithmic framework is entirely different. There, data is represented as vectors, and at each iteration the algorithm identifies the item that is most orthogonal to the current vector, which gets updated with the new item's projection. The limitation of this method is that it can only work in high-dimensional data (e.g., it would not work at all on one-dimensional data). Other work on diverse set selection focused on satisfying fairness constraints while optimizing an additive utility [45]. These methods do not apply to our setting as Max-Min is not additive. Prior work has also examined the satisfaction of fairness constraints or preferences in specialized settings, such as rankings [14, 48, 49]. Work in this domain focuses on specifying and measuring fairness and augmenting ranking algorithms with fairness considerations. Related work on diverse top-k results focuses on returning search results by a combined measure of relevance and dissimilarity to results already produced [5, 42].

Our fairness constraints are based on the definitions of *group fairness* and *statistical parity* [25]. We do not pick a particular definition of fairness, and do not place particular restrictions on the values and distribution of $\langle k_1, \ldots, k_m \rangle$. This model can express equal and proportional representation, as well as any other distribution. There are other, non-parity-based definitions of fairness that fall outside our framework. For example, *individual or causal fairness* [27] examine differences in treatment of individuals from different groups who are otherwise very similar, but these are not the focus of this work.

6 Summary and Future Directions

In this paper, we focused on the problem of diverse data selection under fairness constraints. To the best of our knowledge, our work is the first to introduce fairness constraints to Max-Min diversification. We studied both cases of disjoint and overlapping groups and proposed novel polynomial algorithms with strong approximation guarantees. For the case of disjoint groups, our algorithms have linear running time with respect to the size of the data. Overall, our work augments in significant ways the existing literature of traditional problems that have been studied under group fairness constraints. We discuss here some possible directions that extend our work through the exploration of problem variants, or intuitions towards improvement of the known algorithms and bounds.

Improved bounds. An interesting open question is whether an $\frac{1}{2}$ approximation for Fair Max-Min is possible, as is the case for Max-Min and fair Max-Sum diversification. In Appendix B, we discuss the correspondence between fairness constraints and partition matroids. It is possible that results relevant to matroids can be exploited to improve the algorithms and bounds for the Fair Max-Min problem.

Extending the swap algorithm to the general case. Our Fair-Swap algorithm provides a better bound compared to our Fair-Flow algorithm for the case of $m = 2$ ($\frac{1}{4}$ and $\frac{1}{5}$ respectively). This indicates the possibility that the swap algorithm, if extended to the general case, could perhaps result in a better bound than Fair-Flow.

Problem variants. Our algorithms aim to approximate the diversity score of the optimal solution to Fair Max-Min, while guaranteeing the satisfaction of the fairness constraints. A possible problem variant could explore the relaxation of the fairness constraints, and seek to minimize their violation while guaranteeing a diversity score at least as good as the solution to unconstrained Max-Min diversification. Another interesting future direction is to study the fair variant of other diversity objectives proposed in the literature [16, 32], for which there are currently no known results.

References

1 Zeinab Abbassi, Vahab S. Mirrokni, and Mayur Thakur. Diversity maximization under matroid constraints. In *KDD '13*, pages 32–40, 2013.

2 Pankaj K. Agarwal, Stavros Sintos, and Alex Steiger. Efficient indexes for diverse top-k range queries. In *PODS '20*, page 213–227, 2020.

3 Sepideh Aghamolaei, Majid Farhadi, and Hamid Zarrabi-Zadeh. Diversity maximization via composable coresets. In *CCCG*, 2015.

4 Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *WSDM '09*, page 5–14, 2009.

5 Albert Angel and Nick Koudas. Efficient diversity-aware search. In *SIGMOD '11*, page 781–792, 2011.

6 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.

7 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.

8 Allan Borodin, Aadhar Jain, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions, and dynamic updates. *ACM Trans. Algorithms*, 2017.

9 Allan Borodin, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *PODS '12*, pages 155–166, 2012.

10 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.

11 Matteo Ceccarello, Andrea Pietracaprina, and Geppino Pucci. Fast coreset-based diversity maximization under matroid constraints. In *WSDM '18*, pages 81–89, 2018.

12 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.

13 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.

14 L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. Ranking with fairness constraints. In *ICALP*, 2017.

15 Alfonso Cevallos, Friedrich Eisenbrand, and Rico Zenklusen. Local search for max-sum diversification. In *SODA '17*, page 130–142, 2017.

16 Barun Chandra and Magnús M Halldórsson. Approximation algorithms for dispersion problems. *J. Algorithms*, pages 438–465, 2001.

17 Danny Z. Chen, Jian Li, Hongyu Liang, and Haitao Wang. Matroid and knapsack center problems. *Algorithmica*, pages 27–52, 2016.

18 Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *NIPS'17*, pages 5036–5044, 2017.

19 Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvtiskii. Matroids, matchings, and fairness. In *Proceedings of Machine Learning Research*, PMLR '19, 2019.

20 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.

21 Anesa "Nes" Diaz-Uda, Carmen Medina, and Beth Schill. Diversity's new frontier: Diversity of thought and the future of the workforce. Deloitte Insights, 2013. URL: `https://www2.deloitte.com/us/en/insights/topics/talent/diversitys-new-frontier.html`.

22 M. Drosou and E. Pitoura. Diverse set selection over dynamic data. *IEEE Transactions on Knowledge and Data Engineering*, 26(5):1102–1116, 2014.

23 Marina Drosou, H.V. Jagadish, Evaggelia Pitoura, and Julia Stoyanovich. Diversity in big data: A review. *Big Data*, 5:73–84, 2017.

24 Marina Drosou and Evaggelia Pitoura. Search result diversification. *SIGMOD Rec.*, pages 41–47, 2010.

25 Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *ITCS '12*, pages 214–226, 2012.

26 Erhan Erkut. The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48–60, 1990.

27 Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. Fairness testing: Testing software for discrimination. In *ESEC/FSE '17*, pages 498–510, 2017.

28 Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *WWW '09*, page 381–390, 2009.

29 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985.

30 Refael Hassin, Shlomi Rubinstein, and Arie Tamir. Approximation algorithms for maximum dispersion. *Oper. Res. Lett.*, 21(3):133–137, October 1997.

31 Vivian Hunt, Dennis Layton, and Sara Prince. Why diversity matters. McKinsey & Company, 2015. URL: `https://www.mckinsey.com/business-functions/organization/our-insights/why-diversity-matters`.

32 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.

33 Matthew Jones, Huy Nguyen, and Thy Nguyen. Fair k-centers via maximum matching. In *ICML 2020*, pages 4940–4949, 2020.

34 Matthew Kay, Cynthia Matuszek, and Sean A. Munson. Unequal representation and gender stereotypes in image search results for occupations. In *CHI '15*, page 3819–3828, 2015.

35 Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In *ICML '19*, volume 97, pages 3448–3457, 09–15 June 2019.

36 Michael J. Kuby. Programming models for facility dispersion: The p-dispersion and maxisum dispersion problems. *Geographical Analysis*, 19(4):315–329, 1987.

37 Todd Litman. Evaluating transportation equity: Guidance for incorporating distributional impacts in transportation planning, 2020.

38 Sean A. Munson, Daniel Xiaodan Zhou, and Paul Resnick. Sidelines: An algorithm for increasing diversity in news and opinion aggregators. In *ICWSM*, 2009.

39 Christos Nomikos, Aris Pagourtzis, and Stathis Zachos. Randomized and approximation algorithms for blue-red matching, 2007. `doi:10.1007/978-3-540-74456-6_63`.

40 James B. Orlin. Max flows in o(nm) time, or better. In *STOC'13*, pages 765–774, 2013. `doi:10.1145/2488608.2488705`.

41 James B. Orlin and Xiao-Yue Gong. A fast max flow algorithm. *CoRR*, abs/1910.04848, 2019. `arXiv:1910.04848`.

42 Lu Qin, Jeffrey Xu Yu, and Lijun Chang. Diversifying top-k results. *Proc. VLDB Endow.*, 5(11):1124–1135, July 2012.

43 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.

44 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

45 Julia Stoyanovich, Ke Yang, and H. V. Jagadish. Online set selection with fairness and diversity constraints. In *EDBT*, 2018.

46 Arie Tamir. Obnoxious facility location on graphs. *SIAM J. Discrete Math.*, 4:550–567, November 1991.

47 Yue Wang, Alexandra Meliou, and Gerome Miklau. Rc-index: Diversifying answers to range queries. *Proc. VLDB Endow.*, 11(7):773–786, 2018.

48 Ke Yang, Vasilis Gkatzelis, and Julia Stoyanovich. Balanced ranking with diversity constraints. In *IJCAI'19*, pages 6035–6042, 2019.

49 Ke Yang and Julia Stoyanovich. Measuring fairness in ranked outputs. In *SSDBM '17*, 2017.

50 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 Additional Algorithms and Proofs

In this section, we prove the hardness and approximation bound results for Fair Max-Min, which are formally stated in Corollary 1. Further, we provide proofs for the theoretical results described in Section 4. We also provide the pseudocode for the Fair$^+$-Swap algorithm (Algorithm 5), described in Section 4.1, and the pseudocode for the Fair$^+$-Flow algorithm (Algorithm 6), described in Section 4.2.

Proof of Corollary 1. First, we show that Fair Max-Min is an NP-complete problem. The problem is clearly in NP: If we are given a solution $\mathcal{S}$, we can verify that it satisfies the fairness constraints and compute its diversity score in polynomial time. The unconstrained version of Max-Min diversification is NP-complete [43, 46], and it is a special case of our problem for $m = 1$. Since any instance of Max-Min diversification can be reduced to an instance of Fair Max-Min with $m = 1$, then Fair Max-Min is also NP-complete.

Subsequently, we show that Fair Max-Min cannot be approximated with an approximation factor better than $\frac{1}{2}$. Suppose that there exists a polynomial algorithm that approximates the diversity score of the optimal solution to Fair Max-Min by a factor of $\alpha > \frac{1}{2}$. Then, this algorithm could also solve the unconstrained Max-Min diversification problem with approximation factor α. However, Ravi et al. [43] have shown that unconstrained Max-Min diversification cannot be approximated within a factor better than $\frac{1}{2}$, through a reduction from the clique problem. Therefore, it is not possible for such an algorithm to exist. ◀

Proof of Theorem 7. Let $O = O_{\{1\}} \cup O_{\{2\}} \cup O_{\{1,2\}}$ be the optimal set that maximizes diversity and satisfies the fairness constraints. Let $\ell^*_{\text{fair}} = \text{div}(O)$, which implies that $d(o_1, o_2) \geq \ell^*_{\text{fair}}$ for any pair of optimal elements $o_1, o_2 \in O$. We will show that for any guess $\gamma \leq \ell^*_{\text{fair}}$, Algorithm 5 returns a set $\mathcal{S} = \mathcal{S}_{\{1\}} \cup \mathcal{S}_{\{2\}} \cup \mathcal{S}_{\{1,2\}}$ with $\text{div}(\mathcal{S}) \geq \gamma/4$.

First, note that by the definition of $\mathcal{S}_{\{1,2\}}$ set, it holds that $\text{div}(\mathcal{S}_{\{1,2\}}) \geq \gamma/4$. Next, notice that the $\mathcal{S}^-$ set in line 3 of Algorithm 5 consists of all the points in $X_{\{1,2\}}$, and all single-colored points $< \gamma/4$ apart from some point in $\mathcal{S}_{\{1,2\}}$. As a result, we know that: (1) all the points remaining in $\mathcal{S}^+ = \mathcal{U} \setminus \mathcal{S}^-$ are greater or equal than $\gamma/4$ apart from all the points in $\mathcal{S}_{\{1,2\}}$, and (2) $\mathcal{S}^+$ only contains single-colored points (if there were any bi-colored elements $\geq \gamma/4$ apart from the points in $\mathcal{S}_{\{1,2\}}$, they would have been added to $\mathcal{S}_{\{1,2\}}$).

We further express $\mathcal{S}^+ = \mathcal{S}^+_{\{1\}} \cup \mathcal{S}^+_{\{2\}}$ with $\mathcal{S}^+_{\{i\}} \subseteq X_{\{i\}}$ for $i \in \{1, 2\}$ and $t = |\mathcal{S}_{\{1,2\}}|$. We argue that for any guess $\gamma \leq \ell^*_{\text{fair}}$, $\mathcal{S}^+$ contains at least $k_i - t$ elements for $i \in \{1, 2\}$ that are $\geq \gamma$ apart. Thus, Fair-Swap will be able to find a set of points to satisfy the fairness constraints that are at least $\gamma/4$ apart.

Define $c^-_{\{1\}}, c^-_{\{2\}}, c^-_{\{1,2\}}$ to be the number of optimal points in $O_{\{1\}}, O_{\{2\}}$ and $O_{\{1,2\}}$ present in $\mathcal{S}^-$ and notice that $c^-_{\{1\}} + c^-_{\{2\}} + c^-_{\{1,2\}} \leq t$. This holds because at most one optimal point can be $< \gamma/4$ from a point in $\mathcal{S}_{\{1,2\}}$. Suppose that there exist a pair of optimal points $o_1, o_2 \in O$, and a point $x \in \mathcal{S}_{\{1,2\}}$ such that $d(o_1, x) < \gamma/4$ and $d(o_2, x) < \gamma/4$. Then we derive a contradiction by applying the triangle inequality as: $d(o_1, o_2) \leq d(o_1, x) + d(x, o_2) < \gamma/2 < \ell^*_{\text{fair}}/2$. Consequently, it now follows that $\mathcal{S}^+$ contains at least $k_1 - c^-_{\{1\}} - c^-_{\{1,2\}} \geq k_1 - t$ optimal points of $O_{\{1\}}$, and $k_2 - c^-_{\{2\}} - c^-_{\{1,2\}} \geq k_2 - t$ of $O_{\{2\}}$, which by definition of O are greater or equal than γ apart.

So Fair-Swap will be able to find a set $\mathcal{S}_{\{1\}} \subseteq \mathcal{S}^+_{\{1\}}$ and $\mathcal{S}_{\{2\}} \subseteq \mathcal{S}^+_{\{2\}}$ with the required number of elements such that $\text{div}(\mathcal{S}_{\{1\}} \cup \mathcal{S}_{\{2\}}) \geq \gamma/4$. Thus, we get that $\text{div}(\mathcal{S}) \geq \gamma/4$. If we perform a binary search over all the pairwise distances of the points in $\mathcal{U}$, we will find a guess $\gamma = \ell^*_{\text{fair}}$, which implies the claimed approximation factor for Fair$^+$-Swap. ◀

Algorithm 5 FAIR$^+$-SWAP: Overlapping classes for $m = 2$.

Input: $\mathcal{U}_1, \mathcal{U}_2$: Universe of available elements
$\gamma \in \mathbb{R}$: A guess on the optimum fair diversity
$k_1, k_2 \in \mathbb{Z}_0^+$
Output: at least k_i points in $\mathcal{U}_i$ for $i \in \{1, 2\}$
procedure FAIR$^+$-SWAP
  $\mathcal{S}_{\{1,2\}} \leftarrow$ maximal subset of $X_{\{1,2\}}$ with all points $\geq \gamma/4$ apart
  $\mathcal{S}^- \leftarrow$ all the points in $\mathcal{U}$ that $< \gamma/4$ apart from a point in $\mathcal{S}_{\{1,2\}}$
  $\mathcal{S}^+ \leftarrow \mathcal{U} \setminus \mathcal{S}^-$ ▷$\mathcal{S}^+ = \mathcal{S}^+_{\{1\}} \cup \mathcal{S}^+_{\{2\}} \subseteq X_{\{1\}} \cup X_{\{2\}}$
▷Select the missing points to satisfy the constraints:
  Set $t = |\mathcal{S}_{\{1,2\}}|$
  if $|\mathcal{S}^+ \cap \mathcal{U}_i| \geq k_i - t$ for $i \in \{1, 2\}$ **then**
    $\mathcal{S}_{\{1\}} \cup \mathcal{S}_{\{2\}} \leftarrow$FAIR-SWAP$(\mathcal{S}^+, k_1 - t, k_2 - t)$
    $\mathcal{S} \leftarrow \mathcal{S}_{\{1\}} \cup \mathcal{S}_{\{2\}} \cup \mathcal{S}_{\{1,2\}}$
  else
    $\mathcal{S} \leftarrow \emptyset$ ▷Abort
return $\mathcal{S}$

Proof of Lemma 9. Consider two points $x, y \in C_j$ and let the length of a shortest unweighted path $P_{x,y}$ between x and y in the graph be ℓ. If $\ell \leq M - 1$ then $d(x, y) < (M - 1)d_2$ as required. If $\ell \geq M$ then by Lemma 8, there must exist two points on this path (including end points) in X_L and $X_{L'}$ such that L and L' are *comparable*, i.e., L is a subset of L' or vice versa and this will lead to a contradiction. Consider the subpath $P_{x',y'} \subset P_{x,y}$ such that $x' \in X_L$ and $y' \in X_{L'}$ for some comparable L and L'. If the internal nodes are $x_1, x_2, \ldots$ and these belong to sets $X_{L_1}, X_{L_2}, \ldots$ then by definition of x' and y', the collection of sets $\{L_1, L_2, \ldots, L'\}$ is an anti-chain and hence the size of this collection is at most M by Lemma 8. Hence, the length of the path between x' and y' is also at most M and therefore $d(x', y') < Md_2 = d_1$. But this contradicts $d(x', y') \geq d_1$ because $x' \in X_L$ and $y' \in X_{L'}$ where L and L' are comparable. ◀

Proof of Theorem 11. Note that if the algorithm does not abort then all points are $\geq \gamma/(3M - 1)$ apart since any two points in different connected components are $\geq \gamma/(3M - 1)$ apart. Hence, it remains to argue that if $\gamma \leq \ell^*_{\text{fair}}$ then the algorithm does not abort.

To argue this, we will show it is possible to construct a flow of size $\sum c_L$. And to do this it suffices to, for each $L \subset [m]$, identify c_L different connected components that each include a point from $\cap_{i \in L} \mathcal{U}_i$. Let $O = \bigcup_{L \subset [m]} O_L$ be an optimal solution where $O_L = O \cap X_L$ and let $c_L = |O_L|$. We will henceforth consider the iteration of the algorithm which guessed this set of $\{c_L\}_{L \subset [m]}$ values.

For every point $x \in O$, let $f(x)$ be the closest point in Z where for all i, $x \in \mathcal{U}_i \Rightarrow f(x) \in \mathcal{U}_i$. Note that this requirement ensures that if x is replaced by $f(x)$ then all the fairness constraints are still satisfied. By construction of Z, $d(x, f(x)) < d_1$. Hence, for any $x, y \in O$, $d(x, y) > \ell^*_{\text{fair}} - 2d_1 \geq \gamma - 2\gamma M/(3M - 1) = (M - 1)d_2$, and hence, by Lemma 9, this implies that all points in $f(O)$ are in different connected components. This implies that there exist connected components with the necessary requirements. ◀

B Fairness as a Partition Matroid

While the focus of our work is on fairness constraints in particular, our results apply in general to any type of constraints that can be expressed in terms of a partition matroid. We provide a brief overview of the matroid definition and show that fairness constraints can be expressed as a partition matroid.

Algorithm 6 FAIR$^+$-FLOW: Overlapping classes for $m \geq 3$.

Input: $\mathcal{U}_1, \ldots, \mathcal{U}_m$: Universe of available elements
$c_L \in \mathbb{Z}^+$ for all $L \subset [m]$: A guess of the flow distribution
$\gamma \in \mathbb{R}$: A guess of the optimum fair diversity
$k_1, \ldots, k_m \in \mathbb{Z}_0^+$
Output: at least k_i points in $\mathcal{U}_i$ for $i \in [m]$

procedure FAIR$^+$-FLOW
  Define $d_1 \leftarrow \frac{M\gamma}{3M-1}$ and $d_2 \leftarrow \frac{\gamma}{3M-1}$
  $Z_{[m]} \leftarrow$ maximal subset of $X_{[m]}$ with all points $\geq d_1$ apart
  for $t = m-1, m-2, \ldots, 1$ **do**
    for all sets of L of size t **do**
      $Z_L \leftarrow$ maximal subset of X_L s.t. each point in Z_L is $\geq d_1$ from every other point in

$$Z_L \cup \bigcup_{L' \in [m]: |L'| \geq t+1, L \subset L'} Z_{L'}$$

  $G_Z \leftarrow$ undirected graph with nodes $Z = \bigcup_{L \subset [m]} Z_L$ and edges (z_1, z_2) if $d(z_1, z_2) < d_2$
  $C_1, C_2, \ldots C_t \leftarrow$ Connected components of G_Z
  ▷Construct flow graph
  Construct directed graph $G = (V, E)$ where

$$\begin{aligned} V &= \{a, v_1, \ldots, v_t, b\} \cup \bigcup_{L \subset [m]: |L| > 0} \{u_L\} \\ E &= \{(a, u_L) \text{ with capacity } c_L : \text{non-empty } L \subset [m]\} \\ &\quad \cup \{(v_j, b) \text{ with capacity } 1 : j \in [t]\} \\ &\quad \cup \{(u_L, v_j) \text{ with capacity } 1 : |Z_L \cap C_j| \geq 1\} \end{aligned}$$

  Compute max a-b flow.
  if flow size $< \sum_{L \subset [m]} c_L$ **then return** $\emptyset$ ▷Abort
  else
    $\forall (u_L, v_j)$ with flow add a node in $C_j \in (\cap_{i \in L} \mathcal{U}_i)$ to $\mathcal{S}$
  return $\mathcal{S}$

▶ **Definition 12.** *A matroid $\mathcal{M}$ is a pair $(\mathcal{E}, \mathcal{I})$ where $\mathcal{E}$ is a ground set and $\mathcal{I}$ is a collection of subsets of E (called independent sets). All the independent sets in $\mathcal{I}$ satisfy the following properties:*

- *If $\mathcal{A} \in \mathcal{I}$, then for every subset $\mathcal{B} \subseteq \mathcal{A}$, $\mathcal{B} \in \mathcal{I}$. (Hereditary property)*
- *If $\mathcal{A}, \mathcal{B} \in \mathcal{I}$ with $|\mathcal{A}| > |\mathcal{B}|$, then $\exists e \in \mathcal{A} \setminus \mathcal{B}$ such that $\mathcal{B} \cup \{e\} \in \mathcal{I}$. (Exchange property)*

A maximal independent set in $\mathcal{I}$ (also called a basis for a matroid) is a set for which there is no element outside of the set that can be added so that the set still remains independent. All maximal independent sets of a matroid have equal cardinality which is also called the rank of the matroid, rank($\mathcal{M}$).

▶ **Definition 13.** *A matroid $\mathcal{M} = (\mathcal{E}, \mathcal{I})$ is a partition matroid if $\mathcal{E}$ can be decomposed into m disjoint sets $\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_m$ and $\mathcal{I}$ is defined as $\mathcal{I} = \{S \subseteq \mathcal{E} : |S \cap \mathcal{E}_i| \leq k_i \ \forall \ i \in [m]\}$.*

Note that a maximal independent set (or a basis) for a partition matroid is an independent set that satisfies all the cardinality constraints with equality. For further information on matroids, we refer the interested reader to [44]. Based on the definitions above, in FAIR MAX-MIN the ground set is the universe of elements $\mathcal{U} = \bigcup_{i=1}^{m} \mathcal{U}_i$. Then FAIR MAX-MIN can be expressed as searching for the maximal independent set of the partition matroid defined over $\mathcal{U}$ that maximizes the Max-Min diversity function.

C Results on fair k-center clustering

In this paper, our primary focus has been on fair diversification based on the Max-Min objective. However, as we discussed in Section 1, fair k-center clustering is a closely-related problem. In this section, we formally define k-center clustering, introduce its fair variant and discuss the known approximation results. We then explore how algorithms and intuitions from our work on fair Max-Min diversification can be adapted towards the fair k-center clustering problem to achieve a constant factor 3-approximation.

The k-center and fair k-center clustering problems. The objective of k-center clustering is to identify k cluster centers, such that the maximum distance of any point in the universe of elements $\mathcal{U}$ from its closest cluster center is minimized. This maximum distance is referred to as the *clustering radius*. More formally, given a distance metric d, k-center clustering is expressed by the following minimization problem: $\underset{\mathcal{S}\subseteq\mathcal{U},|\mathcal{S}|=k}{\text{minimize}} \max_{u\in\mathcal{U}} d(u,\mathcal{S})$, where $d(u,\mathcal{S}) = \min_{s\in\mathcal{S}} d(u,s)$. Note that this objective does not preclude cluster centers from being close to each other, and in fact an optimal solution to k-center clustering could be arbitrarily bad for Max-Min diversification.

Algorithms and approximations. Just like Max-Min diversification, k-center clustering is NP-complete. The greedy approximation algorithm proposed by Gonzalez [29] is essentially equivalent to GMM (Algorithm 1) and provides a 2-approximation with linear running time.

Notably, there is recent work that augments the problem with fairness constraints [35]: Given m non-overlapping classes in $\mathcal{U} = \cup_{i=1}^{m}\mathcal{U}_i$ and non-negative integers $\langle k_1,\ldots,k_m\rangle$, the goal is to derive a set of cluster centers $\mathcal{S}$, such that $|\mathcal{S}\cap\mathcal{U}_i| = k_i$. The fair k-center clustering problem can also be expressed by a partition matroid, for which Chen et al. [17] provide a 3-approximation with a quadratic runtime. Kleindessner et al. [35] provide a linear-time 5-approximation algorithm for the case of two classes ($m=2$), and a linear-time $\big(3\cdot 2^{m-1}-1\big)$-approximation for the general case, a result recently improved to $3(1+\epsilon)$ by Chiplunkar et al. [20] and to 3-approximation by Jones et al. [33]. In Section C.1, we adapt the flow algorithm for fair Max-Min diversification, and provide a linear-time 3-approximation for fair k-center clustering. (noting that the three results were derived independently.)

C.1 Fair k-center clustering

We show how we can adapt our Fair-Flow algorithm (Algorithm 3) and design a constant factor 3-approximation for fair k-center clustering with linear running time.

Basic algorithm. We start by presenting a basic algorithm that takes as input a guess γ for the optimum fair clustering radius. If this guess is less than the optimum fair clustering radius r^*_{fair} then the algorithm may abort but otherwise it will return a fair clustering with radius at most 3γ.

Algorithm and intuition. The basic idea behind Fair-Flow-Clust (Algorithm 7) is to construct a set of points $Y=\{y_1,\ldots,y_t\}$ where all distances between these points are $>2\gamma$ apart and all points not in this set are $\leq 2\gamma$ from some point in Y; this can be done via the GMM algorithm (lines 2 and 7). The fact that each pair is $>2\gamma$ apart implies that any k-center clustering, fair or otherwise, with covering radius $\leq\gamma$ has the property that at least one center must be within a distance γ from each y_i and that no center is within distance γ of two points y_i, y_j since, by appealing to the triangle inequality, this would violate the fact that $d(y_i,y_j)>2\gamma$.

Algorithm 7 FAIR-FLOW-CLUST: Fair k-Center Clustering.

Input: $\mathcal{U}_1, \ldots, \mathcal{U}_m$: Universe of available elements
$k_1, \ldots, k_m \in \mathbb{Z}^+$
$\gamma \in \mathbb{R}$: A guess of optimum fair clustering radius.
Output: k_i points in $\mathcal{U}_i$ for each $i \in [m]$

1: **procedure** FAIR-FLOW-CLUST
2: $Y = \{y_1, \ldots, y_{k+1}\} \leftarrow \text{GMM}(\mathcal{U}, \emptyset, k+1)$
3: **for** $j \in [k]$ **do**
4: $D_j \leftarrow \{\text{argmin}_{x \in \mathcal{U}_i} d(x, y_j) : i \in [m]\}$
5: **if** $d(y_{k+1}, \{y_1, \ldots, y_k\}) > 2\gamma$ **then return** $\emptyset$ ▷Abort
6: **else**
7: $Y = \{y_1, \ldots, y_t\}$ with minimum $t \leq k$ such that
$d(y_{t+1}, \{y_1, \ldots, y_t\}) \leq 2\gamma$
8: **for** $j \in [t]$ **do**
9: $C_j \leftarrow \{x \in D_j : d(x, y_j) \leq \gamma\}$
▷Construct flow graph
10: Construct directed graph $G = (V, E)$ where

$$\begin{aligned} V &= \{a, u_1, \ldots, u_m, v_1, \ldots, v_t, b\} \\ E &= \{(a, u_i) \text{ with capacity } k_i : i \in [m]\} \\ &\quad \cup \{(v_j, b) \text{ with capacity } 1 : j \in [t]\} \\ &\quad \cup \{(u_i, v_j) \text{ with capacity } 1 : |Z_i \cap C_j| \geq 1\} \end{aligned}$$

11: Compute max a-b flow.
12: **if** flow size $< t$ **then return** $\emptyset$ ▷Abort
13: **else** ▷max flow is t
14: $\forall (u_i, v_j)$ with flow add a node in C_j with color i to $\mathcal{S}$
return $\mathcal{S}$

The algorithm constructs a sets $C_1, \ldots, C_t$ such that we will be able to argue that if we can pick a fair set of cluster centers from $C_1 \cup \ldots \cup C_t$ such that *exactly* one point is picked in each C_j then we get a clustering with cluster radius 3γ. Furthermore, if $\gamma \geq r^*_{\text{fair}}$, such a set of centers can be proven to exist. We will then be able to find these centers via a reduction to network flow. The network constructed is the same as in Algorithm 3 although the C_j sets in that algorithm are constructed differently. The only difference is that because we need exactly one point in each of $C_1, C_2, \ldots, C_t$, we need to find a flow of size t rather than a flow of size k. Note that if we are able to construct a flow of $t \leq k$, we can arbitrarily add the cluster centers missing from a class $i \in [m]$ without affecting the clustering radius of the solution.

▶ **Theorem 14.** *If $\gamma \geq r^*_{fair}$ then the above algorithm returns a fair clustering with radius at most 3γ. If $\gamma < r^*_{fair}$ then either the algorithm aborts or it returns a fair clustering with radius at most 3γ.*

Proof. Note that if the algorithm does not abort, the algorithm identifies exactly one point in each of the disjoint sets $C_1, \ldots, C_t$ such that at most k_i points of color i are chosen for each color $i \in [m]$. Since the algorithm did not abort at Step 3 we know that all points in $\mathcal{U}$ are within distance 2γ of some point y_i and hence at most distance $2\gamma + \gamma$ from the selected point in C_i. Hence, we return a fair clustering with covering radius at most 3γ as required. It remains to show that if $\gamma \geq r^*_{\text{fair}}$ then the algorithm does not abort. The algorithm does not abort at line 5 since this would imply there exist $k+1$ points that are $> 2\gamma$ from each other and this implies $r^*_{\text{fair}} > \gamma$. Define $E_j = \{x : d(x, y_j) \leq \gamma\}$ and note that the optimum

solution must pick a point in each E_j since otherwise y_j is not covered within distance γ. Hence, we know it is possible to pick at most k_j points of color j such that exactly one point c_j is picked in each E_j. Note that E_j has a point of color i iff C_j has a point of color i. Hence, it is also possible to pick at most k_i points of color i (for each $i \in [m]$) such that exactly one point c_j is picked in each C_j. Hence, there exists a flow of size t where (u_i, v_j) has flow 1 iff c_j has color i and all edges into b are saturated. ◀

Final algorithm. We now proceed as in the case of FAIR-FLOW (Section 3.2): we can either binary search for the good γ over the continuous range $[d_{\min}, d_{\max}]$ or over the discrete set of all distances between points in $Y \cup D_1 \cup D_2 \cup \ldots \cup D_m$. In the first case, we need $O(\log \log_{1+\epsilon} d_{\max}/d_{\min})$ instantiations of the basic algorithm before we find a clustering with approximation ratio $3(1+\epsilon)$. In the second case, we need to sort $O(k^2m^2)$ distances and then need $O(\log k)$ instantiations.

▶ **Theorem 15.** *There is a 3-approximation for fair k-center clustering with running time $O(kn + m^2k^2 \log k)$.*

Proof. Note that Y and $D_1, D_2, \ldots, D_m$ can be computed in $O(kn)$ time. The flow instance has $O(k)$ nodes and $O(mk)$ edges. Hence, it can be solved in $O(mk^2)$ time [40, 41]. The total running time is therefore $O(kn + m^2k^2 \log k + mk^2 \log k)$ as required. ◀