Diverse Data Selection under Fairness Constraints

Zafeiria Moumoulidou Andrew McGregor Alexandra Meliou
College of Information and Computer Sciences
University of Massachusetts Amherst
{zmoumoulidou,mcgregor,ameli}@cs.umass.edu

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, and show empirically that our algorithms perform better than their theoretical bounds.

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, 29, 39], typically defined over numerical attributes (e.g., geographic location, age). On the other hand, fairness is typically defined over categorical attributes (e.g., race, gender) and specifies the appropriate level of representation of different categories or demographic groups, in a set of items. While one could consider combining fairness and diversity into a single objective, comparing numerical and categorical attributes is not straightforward.

Figure 1 demonstrates an example of the principles of diversity and fairness applied to a dataset of Nobel laureates. The first subset of four items 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. Remarkably, the concept of *fair* and *diverse* data selection is also highly motivated by real-world scenarios: *transportation equity* in conjunction with optimizing traditional objectives (e.g., geographic coverage) aims for designing accessible transportation systems for historically disadvantaged groups [34]; formulating *teams* that represent various demographic groups while demonstrating "diversity of thought" is becoming an important hiring goal [19, 20, 30]; 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 [20].



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

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 [20, 39, 42], 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]

[1]Our fairness constraints are based on the definitions of *group fairness* and *statistical parity* [24]. Other definitions that focus on *individual or causal fairness* examine

	Max-Min	Max-Sum
diversification	[29, 39, 42] $\frac{1}{2}$-approximation	
fair diversification	[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] $\left(\frac{1}{2} - \epsilon\right)$-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

Figure 2: 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.

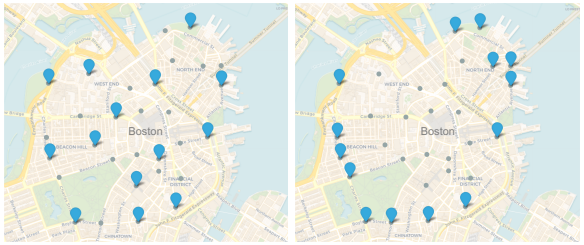


Figure 3: The department of transportation wants to place $k = 14$ new bike sharing stations in downtown Boston among $n = 30$ candidate locations. (Left): Max-Min selects locations that geographically *cover* downtown. (Right): Max-Sum selects locations on the outskirts of downtown.

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, 32], ranking systems [14, 44–46] and set selection [41]. We proceed to review prior work in closely-related problems and describe how our contributions augment the existing literature. (Summary shown in Figure 2.)

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, 25, 29, 33, 39]. Similar formulations have also been studied in the context of obnoxious facility location on graphs [42]. 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 3). 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, 39].

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. [2] 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 Max-Min diversification. To the best of our knowledge, this paper is the first to introduce fairness constraints to Max-Min diversification, and thus there are no prior results that we know of on this problem. In contrast to Max-Sum, which is an additive objective, Max-Min is harder to analyze. In the case of non-overlapping groups, we propose two polynomial-time algorithms: a swap-based greedy algorithm that achieves a $\frac{1}{4}$-approximation for $m = 2$, and a max-flow-based algorithm that achieves a $\frac{1}{3m-1}$-approximation for any m. Additionally, we show that for constant m and $k = o(\log n)$, we can achieve a constant $\frac{1}{5}$-approximation. In the generalized setting, when the groups are allowed to overlap, we extend our algorithms to provide a $\frac{1}{4}$-approximation guarantee for $m = 2$, and a $\frac{1}{3\binom{m}{\lfloor m/2 \rfloor}-1}$-approximation guarantee for any m.

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 will tend to be far from each other, and thus k-center clustering can serve as another mechanism to perform diverse data

differences in treatment of individuals from different groups who are otherwise very similar, but these are not the focus of this work.

[2] Section 3.2 describes the correspondence between group fairness constraints and partition matroids.



Figure 4: 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 $\{1, 6\}$.

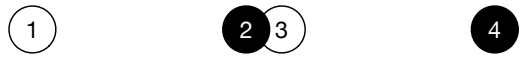


Figure 5: An optimal solution for the clustering problem may be arbitrarily bad for the diversity problem. Suppose we have to pick one white and one black point. Set $\{2, 3\}$ is an optimal solution for clustering but yields an arbitrarily bad approximation ratio for the diversity problem as points 2 and 3 can be arbitrarily close together.

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 [28] for the clustering problem and by Ravi et al. [39] and Tamir [42] 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. [32] 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 4). Furthermore, we note that an optimal solution for fair k-center clustering can be arbitrarily bad for the Max-Min objective (e.g., Figure 5). 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. [32] provide a linear-time algorithm with a $\left(3 \cdot 2^{m-1} - 1\right)$-approximation, while more recent work improved this bound to $3(1 + \epsilon)$ [**?**]. In our Appendix, we also provide a slight improvement to this bound, with a linear-time algorithm with constant 3-factor approximation.

Outline of contributions. In this paper, we study the problem of diverse data selection under fairness constraints. 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 (Section 2), we introduce and formally define the problem of *fair* Max-Min diversification focusing on non-overlapping groups. To the best of our knowledge, no prior work has studied the Max-Min diversification objective under fairness constraints. We further discuss its complexity and approximability, and describe how our algorithmic frameworks support any constraints that can be expressed in terms of partition matroids (Section 3).

Algorithm 1 GMM Algorithm

Input: $\mathcal{U}$: Universe of available elements
$k \in \mathbb{Z}^+$
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**
    $S \leftarrow$ a randomly chosen point in $\mathcal{U}$
  while $|\mathcal{S}| < k$ **do**
    $x \leftarrow \operatorname{argmax}_{u \in \mathcal{U}} \min_{s \in \mathcal{S} \cup I} d(u, s)$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{x\}$
  return $\mathcal{S}$

- 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 4.1).
- 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 4.2). We further demonstrate that for constant m and $k = o(\log n)$, we can achieve a constant $\frac{1}{5}$-approximation, also in linear time (Section 4.3).
- 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 5).
- We demonstrate that the empirical approximation performance of our algorithms is typically much better than their theoretical bounds (Section 6). Our Appendix contains further experimental results over real-world datasets that demonstrate that the diversity loss from enforcing fairness is usually small.

2 BACKGROUND AND PRELIMINARIES

In this section, we review necessary background and preliminaries on diverse data selection, the Max-Min diversification objective and relevant approximations. We also introduce the notation used in this paper. We start by focusing on the case of *non-overlapping* groups; we will generalize the problem statement and algorithms to overlapping groups in Section 5.

Diverse data selection. We assume a universe of elements $\mathcal{U}$ of size n and a positive integer $k \leq |\mathcal{U}|$. Diverse data selection seeks to identify a set $\mathcal{S} \subseteq \mathcal{U}$ and $|\mathcal{S}| = k$, such that the diversity of $\mathcal{S}$ is maximized.

Fair and diverse data selection. 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 non-negative integers $\langle k_1, \ldots, k_m \rangle$. Fair and diverse data selection seeks to identify a set $\mathcal{S} \subseteq \mathcal{U}$ such that for all i, $|\mathcal{S} \cap \mathcal{U}_i| = k_i$, and the diversity of $\mathcal{S}$ is maximized.

For both problems, 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. Our algorithms (for the non-overlapping case) have linear-time complexity with respect to n, and we will simply refer to those as linear-time.

Max-Min Diversification. 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 $\mathcal{S}$. More formally, let $d : \mathcal{U} \times \mathcal{U} \to \mathbb{R}_0^+$ be a distance function, such that $d(u, v)$ captures the dissimilarity of the elements $u, v \in \mathcal{U}$. Then, the Max-Min diversity score of a set $\mathcal{S}$ is

$$\text{div}(\mathcal{S}) = \min_{\substack{u,v \in \mathcal{S} \\ u \neq v}} d(u, v)$$

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.

Algorithms and approximations. This problem formulation was initially studied in the operation research literature by Ravi et al. [39] and in the context of facility location on graphs by Tamir [42]. 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. [39] 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. [39] and Tamir [42] 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., [32, 43] for details.

Pseudometric assumption. While the Max-Min diversity objective is defined over general distance functions, in this work, we focus on pseudometric distance functions. A distance function defined as $d : \mathcal{U} \times \mathcal{U} \to \mathbb{R}_0^+$ is a pseudometric if it satisfies the following properties for every $u, v \in \mathcal{U}$:

$$d(u, u) = 0$$
$$d(u, v) = d(v, u) \text{ (symmetry)}$$
$$d(u, v) \leq d(u, w) + d(w, v) \text{ (triangle inequality)}$$

Our contributions to this problem. In Section 3, we augment Max-Min diversification with fairness constraints; to the best of our knowledge, this is the first paper to define and investigate this problem under the Max-Min objective. In Section 4, we provide linear-time algorithms with strong approximation guarantees for this problem in the case of non-overlapping groups. In Section 5, we extend our results to design polynomial-time algorithms with strong approximation guarantees for the generalized setting of overlapping groups.

Notation	Description
$\mathcal{U}$	Universe of elements
n	# elements in $\mathcal{U}$
k	# elements desired in data selection
m	# classes / demographic groups in $\mathcal{U}$
k_i	# elements desired from group i
$\mathcal{S}$	Subset of $\mathcal{U}$ of size k
$d(\cdot, \cdot)$	Pseudometric distance function
$\text{div}(\mathcal{S})$	Minimum pairwise distance of elements in $\mathcal{S}$
GMM	Algorithm 1: Greedy approximation of Max-Min diversification with $\frac{1}{2}$-approximation guarantee

Figure 6: Summary of notations used in the paper.

3 FAIR MAX-MIN DIVERSIFICATION

In this section, we formally define the Max-Min diversification problem under fairness constraints which is a generalization of Max-Min diversification. We characterize the hardness of the problem, and the hardness of its approximation, and describe its connection to partition matroids.

3.1 Problem 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:

Fair Max-Min:

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

Intuitively, Fair Max-Min aims to derive the set with the maximum diversity score $\text{div}(\mathcal{S})$, while satisfying the fairness constraints. 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 $m = 1$.

Theorem 1 (Hardness). *Determining if there exists a solution to Fair Max-Min with diversity score $\geq \delta$ is NP-complete.*

Proof. 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 [39, 42], 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. □

Theorem 2 (Approximability Bound). *There exists no polynomial-time α-approximation algorithm for Fair Max-Min with $\alpha > \frac{1}{2}$, unless P=NP.*

Proof. 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. [39] 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. □

The main contributions of our work focus on providing approximation algorithms for Fair Max-Min, assuming non-overlapping groups, whose running time is linear in n and have strong theoretical approximation guarantees. We present our algorithmic and theoretical contributions in Section 4. In summary, we present an algorithm with a $\frac{1}{4}$ approximation bound for the case of $m = 2$, and a second algorithm with a $\frac{1}{3m-1}$ approximation bound for the general case. Finally, we present a simple algorithm that offers a constant $\frac{1}{5}$ approximation bound for the case of small m and k. In Section 5, we extend our results and present polynomial-time algorithms for the case of overlapping groups.

3.2 Fairness as a Partition Matroid

While our focus 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.

Definition 1. *A matroid $\mathcal{M}$ is a pair $(\mathcal{E}, \mathcal{I})$ where $\mathcal{E}$ is a ground set and $\mathcal{I}$ is the set of independent sets defined as the union of any subset $\mathcal{I} \subseteq 2^{\mathcal{E}}$. 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 2. *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 $\mathcal{I} = \{S \subseteq \mathcal{E} : |S \cap \mathcal{E}_i| \leq k_i \ \forall\, i \in [1, m]\}$.*

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 [**?**].

Based on the above definitions, 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.

4 APPROXIMATING DIVERSITY

In Section 3.1, we showed that the *fair* formulation for the Max-Min diversification problem is NP-hard so we can only hope for designing an approximate solution to it. The algorithm framework we propose uses GMM (Algorithm 1) as a building block. We start with a $\frac{1}{4}$ approximation when $m = 2$. Then we present 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 will grow exponentially in k when $k = \Omega(\log n)$.

Throughout the next two sections, we will refer to each of the m groups as different colors. We think this makes the exposition easier to follow.

Algorithm 2 Fair-Swap: Fair Diversification for $m = 2$

Input: $\mathcal{U}_1, \mathcal{U}_2$: Set of points of color 1 and 2
$k_1, k_2 \in \mathbb{Z}^+$
Output: k_i points in $\mathcal{U}_i$ for $i \in \{1, 2\}$
1: **procedure** Fair-Swap
▷Color-Blind Phase:
2: $\mathcal{S} \leftarrow \text{GMM}(\mathcal{U}, \emptyset, k)$
3: $\mathcal{S}_i = \mathcal{S} \cap \mathcal{U}_i$ for $i \in \{1, 2\}$

▷Balancing Phase:
4: Set $U = \text{argmin}_i(k_i - |\mathcal{S}_i|)$ ▷Under-satisfied set
5: $O = 3 - U$ ▷Over-satisfied set
6: Compute:
$E \leftarrow \text{GMM}(\mathcal{U}_U, \mathcal{S}_U, k_U - |\mathcal{S}_U|)$
$R \leftarrow \{\text{argmin}_{x \in \mathcal{S}_O} d(x, e) : e \in E\}$
return $(\mathcal{S}_U \cup E) \cup (\mathcal{S}_O \setminus R)$

4.1 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 and intuition (Fair-Swap). Algorithm 2 (Fair-Swap) 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 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

$$\text{div}(\mathcal{S}) \geq \ell^*/2 \geq \ell^*_{\text{fair}}/2 \,. \tag{1}$$

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

$$\text{div}(\mathcal{S}_U \cup E) \geq \ell^*_{\text{fair}}/2 \,. \tag{2}$$

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 $< k - |\mathcal{S}_U|$ 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.

From, Eq. 1 and Eq. 2 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 closets points the distance between all pairs is $\geq \ell^*_{\text{fair}}/4$ as required.

We summarize the analysis of this section as follows:

Theorem 3. *Algorithm 2 is a* $1/4$*-approximation algorithm for the fair diversification problem when* $m = 2$ *that runs in time* $O(kn)$.

4.2 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 (Fair-Flow). The approach of 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 7 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
$k_1, \ldots, k_m \in \mathbb{Z}^+$
$\gamma \in \mathbb{R}$: 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**
    $Y_i \leftarrow \text{GMM}(\mathcal{U}_i, \emptyset, k)$
  $Z_i \leftarrow$ 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 $\mathcal{S}$

Lemma 1. *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.*

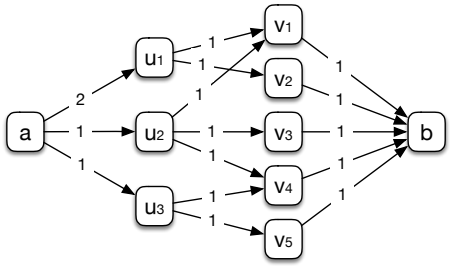


Figure 7: The graph construction in Algorithm 3 (line 7) corresponding to $m = 3$, $k_1 = 2$, $k_2 = 1$, $k_3 = 1$. Points of color 1 are contained in C_1 and C_2. Points of color 2 are contained in C_1, C_3, and C_4. Points of color 3 are contained in C_4 and C_5. Note there is an a-b flow of size $k_1 + k_2 + k_3$ 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 $i \in [m]$.

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 Y_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 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 1, 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

$$d_{\min} = \min_{x,y \in X : x \neq y} d(x,y) \text{ and } d_{\max} = \max_{x,y \in X} d(x,y) .$$

Hence, there exists a guess $\gamma = (1+\epsilon)^i d_{\min}$ for some

$$i \in \{0, 1, 2, \ldots, \lceil \log_{1+\epsilon} R \rceil\} \text{ where } R := d_{\max}/d_{\min}$$

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

$$O(\log(1 + \lceil \log_{1+\epsilon} R \rceil) = O(\log(\epsilon^{-1}) + \log\log R) .$$

This analysis can be slightly improved by noting ℓ^*_{fair} is at most twice the minimum distance amongst $Y_1, \ldots, Y_m$.

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 from the previous subsection.

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 [36, 37]. Note that the binary search requires us to construct and solve $O(\log(km))$ flow instances. Hence the total running time is as claimed. □

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 $\frac{1}{(3m-1)(1+\epsilon)}$.

Algorithm 4 FAIR-GMM: Fair Diversification for small k

Input: $\mathcal{U}_1, \ldots, \mathcal{U}_m$: Universe of available elements
$k_1, \ldots, k_m \in \mathbb{Z}^+$
Output: k_i points in $\mathcal{U}_i$ for $i \in [m]$
1: **procedure** FAIR-GMM
2: **for** $i \in [m]$ **do**
3: $Y_i \leftarrow \text{GMM}(\mathcal{U}_i, \emptyset, k)$
4: 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.

4.3 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 (FAIR-GMM). The basic approach of 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:



Figure 8: 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 all i that are critical, let $\mathcal{S}_i = f(O_i)$ and let $D = \cup_{i:\text{critical}} \mathcal{S}_i$. Note that $\text{div}(D) > \ell^*_{\text{fair}} - 4\ell^*_{\text{fair}}/5 = \ell^*_{\text{fair}}/5$.
- 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. □

5 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

$$X_L = \Big(\bigcap_{i \in L} \mathcal{U}_i\Big) \cap \Big(\bigcup_{j \notin L} \mathcal{U}_j\Big)$$

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 8). Formally, we define the problem as follows:

FAIR$^+$ MAX-MIN:

$$\begin{array}{ll} \underset{\mathcal{S} \subseteq \mathcal{U}}{\text{maximize}} & \min\limits_{\substack{u,v \in \mathcal{S} \\ u \neq v}} d(u,v) \\ \text{subject to} & |\mathcal{S} \cap \mathcal{U}_i| \geq k_i, \ \forall\, i \in [m] \end{array}$$

5.1 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 (Fair$^+$-Swap). The algorithm takes as input a guess γ for the optimum fair diversity. We show that if $\gamma \le \ell^*_{\text{fair}}$, 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 $\text{div}(\mathcal{S}) \ge \gamma/4$ (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 $\ge \gamma$ 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 6) 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 C.

5.2 Fair-and-Diverse Selection (Overlaps): $m \ge 3$

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.

5.2.1 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:

$$\begin{aligned}\mathcal{U}_1 &= X_{\{1\}} \cup X_{\{1,2\}} \cup X_{\{1,3\}} \cup X_{\{1,2,3\}}\\ \mathcal{U}_2 &= X_{\{2\}} \cup X_{\{1,2\}} \cup X_{\{2,3\}} \cup X_{\{1,2,3\}}\\ \mathcal{U}_3 &= X_{\{3\}} \cup X_{\{1,3\}} \cup X_{\{2,3\}} \cup X_{\{1,2,3\}}\ .\end{aligned}$$

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 $Z_{\{1,2,3\}} \subset X_{\{1,2,3\}}$ subject to the constraint that all points are at least d_1 from each other. 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 7 (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 2 (Sperner's Lemma). *A collection of sets is called an* antichain *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 this distance is $< d_2$. The new algorithm proceeds similarly but with new parameters:

$$d_1 \leftarrow \frac{M\gamma}{3M-1} \qquad \text{and} \qquad d_2 \leftarrow \frac{\gamma}{3M-1}\ .$$

With this setting of the parameters and appealing to Lemma 2 we prove an upper bound on the distance between any two points in the same connected components.

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 a, b such that $a \in X_L$ and $b \in X_{L'}$ where $L \subset L'$.

We provide the proof of Lemma 3 in Appendix C.

5.2.2 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

[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 $d(x, y) < d_1$.

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 1. *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.

Theorem 8. *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.*

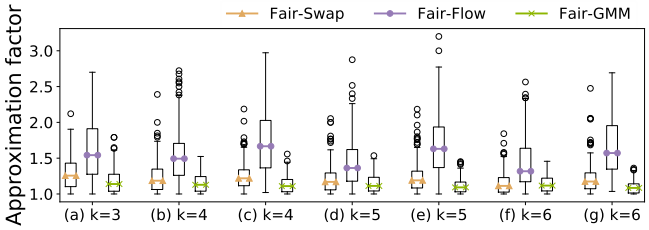
We provide the proof of Theorem 8 in Appendix C. The rest of the algorithm and analysis follows similarly as Algorithm 3, where we binary search for γ in either a continuous or discrete space. The running time is increased by a factor of $k^{2^m - m - 1}$ because of the need to guess the values $\{c_L\}_{L \subset [m]}$; thus Fair$^+$-Flow is a polynomial-time algorithm with a $\frac{1}{3\binom{m}{\lfloor m/2 \rfloor}}$-approximation guarantee.

6 EXPERIMENTAL EVALUATION

The contributions of our work are mostly of theoretical nature. In this section, we provide some experimental results to evaluate the behavior of our algorithms in terms of their approximation performance, using synthetic data with computable and known optimal solution. For ease of presentation and readability, we always present the approximation ratio as a value > 1 (i.e., an approximation with diversity score $\frac{1}{4}$ of the optimal will be reported as 4). In the Appendix, we include further experiments on real-world datasets to evaluate the *price of fairness*, namely, the diversity loss that we incur by enforcing fairness constraints.

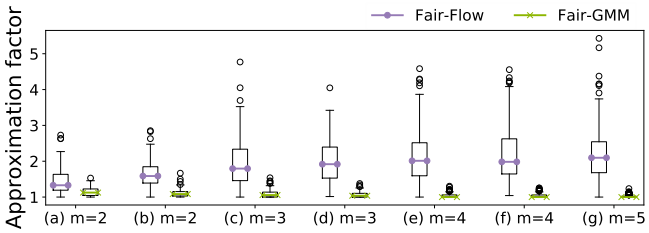
Fair-Swap vs Fair-Flow vs Fair-GMM ($m = 2$). Figure 9 shows an empirical comparison of the approximation quality (OPT/ALG) of our three algorithms, Fair-Swap (Algorithm 2), Fair-Flow (Algorithm 3), and Fair-GMM (Algorithm 4), for the case of $m = 2$. For this experiment we vary k over a small dataset of $n = 50$ elements. The small data size allows us to compute the true optimal solution via exhaustive search. For the synthetic data generation, we follow the process described in the work by Kleindessner et al. [32]. In particular, the dataset consists of $n = 50$ vertices, randomly-connected according to the Erdős-Rényi model: there is an edge between each pair of nodes with probability $p = 2 \log n/n$ and the final graph is connected with high probability. Then, we randomly and uniformly assign each edge a weight $w \in [0, 100]$ while the pseudometric

[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.



	(a)	(b)	(c)	(d)	(e)	(f)	(g)
$\langle \mathbf{k_1}, \mathbf{k_2} \rangle$	$\langle 2, 1 \rangle$	$\langle 3, 1 \rangle$	$\langle 2, 2 \rangle$	$\langle 4, 1 \rangle$	$\langle 3, 2 \rangle$	$\langle 5, 1 \rangle$	$\langle 3, 3 \rangle$

Figure 9: Comparison of the approximation performance of Fair-Swap, Fair-Flow, and Fair-GMM over a small synthetic dataset ($n = 50$) with computable optimal solution for $m = 2$.



	(a)	(b)	(c)	(d)	(e)	(f)	(g)
$\langle \mathbf{k_1}, \ldots \mathbf{k_m} \rangle$	$\langle 5, 1 \rangle$	$\langle 3, 3 \rangle$	$\langle 4, 1, 1 \rangle$	$\langle 3, 2, 1 \rangle$	$\langle 3, 1, 1, 1 \rangle$	$\langle 2, 2, 1, 1 \rangle$	$\langle 2, 1, 1, 1, 1 \rangle$

Figure 10: Comparison of the approximation performance of Fair-Flow and Fair-GMM over a small synthetic dataset ($n = 50$) with computable optimal solution for $m \geq 2$.

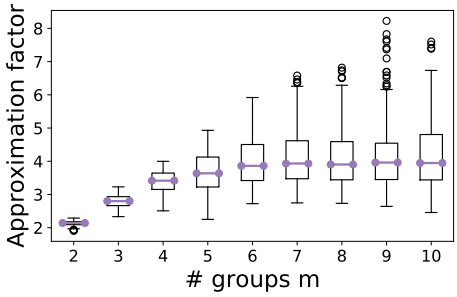


Figure 11: Approximation performance of Fair-Flow as m grows over a synthetic dataset with known optimal solution. ($n = 10,128$ and $k = 128$)

distance used in this setting is the shortest path between a pair of nodes. The partitions of the data are generated by randomly assigning each vertex to one of the m available groups. We report results over 200 iterations of the experiment.

We find that, in practice, all algorithms perform better than their theoretical guarantees (4, 5, and 5, for Fair-Swap, Fair-Flow and Fair-GMM, respectively). Fair-Swap consistently outperforms Fair-Flow, while Fair-GMM has the overall best results. Fair-GMM, despite its restrictive nature, explores the feasible solution space more broadly since it independently optimizes each group and then *exhaustively* searches for the best fair subset. For example, note that the solution derived by Fair-Flow is guaranteed to be explored by Fair-GMM as well: both algorithms build a solution by selecting elements within the $Y = \bigcup_{i=1}^{m} Y_i$ set, where Y_i is a diverse set of at most k elements of color i picked by GMM.

Fair-Flow vs Fair-GMM ($m \geq 2$). We use the same dataset generation process (with $n = 50$ and $k = 6$) to compare the performance of Fair-Flow and Fair-GMM as m increases. Figure 10 reports the results over 200 iterations of the experiment. As expected, the approximation performance for Fair-Flow worsens as m increases, albeit not drastically.

We further evaluate the performance of Fair-Flow over larger datasets with a known optimal solution using the following process: Given an integer k, we construct a unit hypercube in the D-dimensional space, where $D = \log_2 k$. Note that the unit hypercube has $k = 2^D$ *corners* that are at least one unit apart from one another; thus, this set of k elements has a diversity score equal to 1. We now add any number of points inside the hypercube, by uniformly picking each dimension out of the D to be in the $(0, 1)$ range; any of these points will lead to diversity score less than 1, thus the optimal set of size k is guaranteed to be the corners. The partitions of the data are generated by randomly assigning each point to one of m possible groups. Subsequently, we set the fairness constraints $\langle k_1, \ldots, k_m \rangle$ to correspond to the group assignment of the corner points, so as to preserve the known optimal solution.

Figure 11 shows the approximation behavior of Fair-Flow, as m increases, for deriving a set of size $k = 128$ over a dataset of $n = 10,128$ elements. (Fair-GMM is not included in this experiments, because due to its restrictions on k and m it is no longer practical.). As expected, the approximation worsens as m increases, but remains well below the theoretical bounds. For example, the theoretical bound of Fair-Flow for $m = 10$ is 29, but empirically, the median ratio is around 4.

7 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, 35] 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 [20, 21].

Max-Min and Max-Sum diversification are two of the most well studied distance-based models[16, 27, 29, 39], 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). In the streaming and distributed settings, prior work uses the idea of (composable) core-sets that have been proven to be a useful theoretical tool for designing online and distributed algorithms.

Indyk et al. [31] design (composable) core-sets for various diversity models in general metric spaces, including Max-Min and Max-Sum. Some of these results are further improved in the work by Aghamolaei et al. [3], resulting in the currently best known results in metric spaces: $\frac{1}{3}$ and $\left(\frac{1}{6} - \epsilon\right)$-approximate core-sets for Max-Min and Max-Sum respectively. Ceccarello et al. [12] study the diversification problem in metric spaces of bounded doubling dimension and design $(1 - \epsilon)$-approximate core-sets for both Max-Min and Max-Sum diversity models among others. In a recent work, Borassi et al. [7] design novel algorithms in the sliding window setting. There is also a line of work that focuses on designing efficient indexing schemes for result diversification [2, 22, 43], which however is not the focus of our work.

There is relatively little prior work on constrained diversification. The closest to our work is fair Max-Sum diversification (discussed in Sections 1 and 2) and fair k-center clustering (discussed in Section 1 and Appendix B). To the best of our knowledge, our work is the first to augment the traditional Max-Min 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 [41]. 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, 44, 45]. 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, 38].

Our fairness constraints are based on the definitions of *group fairness* and *statistical parity* [24]. 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* [26] examine differences in treatment of individuals from different groups who are otherwise very similar, but these are not the focus of this work.

8 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.

Extending the swap algorithm to the general case. Our FairSwap algorithm provides a better bound compared to our FairFlow 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.

Using partition matroids. As we discussed in Section 3.2, our fairness constraints correspond to partition matroids, and in fact our results apply to the more general case of constraints that can be expressed as 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.

Approximating fairness. 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.

REFERENCES

[1] Zeinab Abbassi, Vahab S. Mirrokni, and Mayur Thakur. 2013. Diversity Maximization Under Matroid Constraints. In *KDD '13*. 32–40.

[2] Pankaj K. Agarwal, Stavros Sintos, and Alex Steiger. 2020. Efficient Indexes for Diverse Top-k Range Queries. In *PODS '20*. 213âĂŞ227.

[3] Sepideh Aghamolaei, Majid Farhadi, and Hamid Zarrabi-Zadeh. 2015. Diversity Maximization via Composable Coresets. In *CCCG*.

[4] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying Search Results. In *WSDM '09*. 5âĂŞ14.

[5] Albert Angel and Nick Koudas. 2011. Efficient Diversity-Aware Search. In *SIGMOD âĂŹ11*. 781âĂŞ792.

[6] Aditya Bhaskara, Mehrdad Ghadiri, Vahab Mirrokni, and Ola Svensson. 2016. Linear Relaxations for Finding Diverse Elements in Metric Spaces. In *NIPSâĂŹ16*. 4105âĂŞ4113.

[7] Michele Borassi, Alessandro Epasto, Silvio Lattanzi, Sergei Vassilvitskii, and Morteza Zadimoghaddam. 2019. Better Sliding Window Algorithms to Maximize Subadditive and Diversity Objectives. In *PODS '19*. 254âĂŞ268.

[8] Allan Borodin, Aadhar Jain, Hyun Chul Lee, and Yuli Ye. 2017. Max-Sum Diversification, Monotone Submodular Functions, and Dynamic Updates. *ACM Trans. Algorithms* (2017).

[9] Allan Borodin, Hyun Chul Lee, and Yuli Ye. 2012. Max-Sum Diversification, Monotone Submodular Functions and Dynamic Updates. In *PODS '12*. 155–166.

[10] Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *SIGIR âĂŹ98*. 335âĂŞ336.

[11] Matteo Ceccarello, Andrea Pietracaprina, and Geppino Pucci. 2018. Fast Coreset-based Diversity Maximization Under Matroid Constraints. In *WSDM '18*. 81–89.

[12] Matteo Ceccarello, Andrea Pietracaprina, Geppino Pucci, and Eli Upfal. 2017. MapReduce and Streaming Algorithms for Diversity Maximization in Metric Spaces of Bounded Doubling Dimension. *Proc. VLDB Endow.* (2017), 469âĂŞ480.

[13] Elisa Celis, Vijay Keswani, Damian Straszak, Amit Deshpande, Tarun Kathuria, and Nisheeth Vishnoi. 2018. Fair and Diverse DPP-Based Data Summarization. In *ICML '2018*. 716–725.

[14] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. 2017. Ranking with Fairness Constraints. In *ICALP*.

[15] Alfonso Cevallos, Friedrich Eisenbrand, and Rico Zenklusen. 2017. Local Search for Max-Sum Diversification. In *SODA âĂŹ17*. 130âĂŞ142.

[16] Barun Chandra and Magnús M Halldórsson. 2001. Approximation Algorithms for Dispersion Problems. *J. Algorithms* (2001), 438–465.

[17] Danny Z. Chen, Jian Li, Hongyu Liang, and Haitao Wang. 2016. Matroid and Knapsack Center Problems. *Algorithmica* (2016), 27–52.

[18] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2017. Fair Clustering Through Fairlets. In *NIPS'17*. 5036–5044.

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

[20] Marina Drosou, H.V. Jagadish, Evaggelia Pitoura, and Julia Stoyanovich. 2017. Diversity in Big Data: A Review. *Big Data* 5 (2017), 73–84.

[21] Marina Drosou and Evaggelia Pitoura. 2010. Search Result Diversification. *SIGMOD Rec.* 1 (2010), 41–47.

[22] M. Drosou and E. Pitoura. 2014. Diverse Set Selection Over Dynamic Data. *IEEE Transactions on Knowledge and Data Engineering* 26, 5 (2014), 1102–1116.

[23] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/datasets/Adult

[24] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness Through Awareness. In *ITCS '12*. 214–226.

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

[26] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. 2017. Fairness Testing: Testing Software for Discrimination. In *ESEC/FSE '17*. 498–510.

[27] Sreenivas Gollapudi and Aneesh Sharma. 2009. An Axiomatic Approach for Result Diversification. In *WWW âĂŹ09*. 381âĂŞ390.

[28] Teofilo F. Gonzalez. 1985. Clustering to Minimize the Maximum Intercluster Distance. *Theor. Comput. Sci.* 38 (1985), 293–306.

[29] Refael Hassin, Shlomi Rubinstein, and Arie Tamir. 1997. Approximation Algorithms for Maximum Dispersion. *Oper. Res. Lett.* 21, 3 (Oct. 1997), 133–137.

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

[31] Piotr Indyk, Sepideh Mahabadi, Mohammad Mahdian, and Vahab S. Mirrokni. 2014. Composable Core-Sets for Diversity and Coverage Maximization. In *PODS âĂŹ14*. 100âĂŞ108.

[32] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. 2019. Fair k-Center Clustering for Data Summarization. In *ICML '19*, Vol. 97. 3448–3457.

[33] Michael J. Kuby. 1987. Programming Models for Facility Dispersion: The p-Dispersion and Maxisum Dispersion Problems. *Geographical Analysis* 19, 4 (1987), 315–329.

[34] Todd Litman. 2020. Evaluating Transportation Equity: Guidance For Incorporating Distributional Impacts in Transportation Planning.

[35] Sean A. Munson, Daniel Xiaodan Zhou, and Paul Resnick. 2009. Sidelines: An Algorithm for Increasing Diversity in News and Opinion Aggregators. In *ICWSM*.

[36] James B. Orlin. 2013. Max flows in O(nm) time, or better. In *STOC'13*. 765–774.

[37] James B. Orlin and Xiao-Yue Gong. 2019. A Fast Max Flow Algorithm. *CoRR* abs/1910.04848 (2019). arXiv:1910.04848

[38] Lu Qin, Jeffrey Xu Yu, and Lijun Chang. 2012. Diversifying Top-k Results. *Proc. VLDB Endow.* 5, 11 (July 2012), 1124âĂŞ1135.

[39] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. 1994. Heuristic and Special Case Algorithms for Dispersion Problems. *Oper. Res.* 42, 2 (April 1994), 299–310.

[40] P. Cortez S. Moro and P. Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* 62 (2014), 22 – 31.

[41] Julia Stoyanovich, Ke Yang, and H. V. Jagadish. 2018. Online Set Selection with Fairness and Diversity Constraints. In *EDBT*.

[42] Arie Tamir. 1991. Obnoxious Facility Location on Graphs. *SIAM J. Discrete Math.* 4 (11 1991), 550–567.

[43] Yue Wang, Alexandra Meliou, and Gerome Miklau. 2018. RC-Index: Diversifying Answers to Range Queries. *Proc. VLDB Endow.* 11, 7 (2018), 773âĂŞ786.

[44] Ke Yang, Vasilis Gkatzelis, and Julia Stoyanovich. 2019. Balanced Ranking with Diversity Constraints. In *IJCAI'19*. 6035–6042.

[45] Ke Yang and Julia Stoyanovich. 2017. Measuring Fairness in Ranked Outputs. In *SSDBM âĂŹ17*.

[46] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. FA*IR: A Fair Top-k Ranking Algorithm. In *CIKM '17*. 1569–1578.

APPENDIX

A EVALUATION: PRICE OF FAIRNESS

We provide some additional experiments with two real-world datasets to examine the *price of fairness*, namely how much diversity loss do we incur by enforcing fairness constraints, compared to the unconstrained version of the diversification problem. Further, we measure how much a fairness-agnostic solution violates the fairness constraints.

In particular, we compare the diversity score of sets derived by Fair-Flow (Algorithm 3) against the ones derived by the fairness-agnostic GMM (Algorithm 1). To model how unfair a GMM solution can be, we observe that the fairness constraints define a *target* distribution across the m different groups within a fair set. So, an intuitive way to model unfairness, or equivalently the degree by which a GMM solution violates the constraints, is as the distance between the *target* distribution, that a Fair-Flow solution follows, and the distribution that a GMM solution follows.

Specifically, we use the Jensen-Shannon distance (JS-distance) defined as the square root of Jensen-Shannon divergence, a symmetric version of KL-divergence which is a frequently used to measure dissimilarity between two distributions. The JS-distance is a metric that takes values within $[0, 1]$ with higher values indicating lower similarity (e.g the JS-distance of two identical distributions is equal to zero). Given a pair of distributions p, q, the JS-distance is defined as:

$$D_{JS}(p, q) = \sqrt{\frac{D_{KL}(p||z)+D_{KL}(q||z)}{2}}$$

where z is the pointwise mean of p, q and D_{KL} is the KL-divergence, defined as:

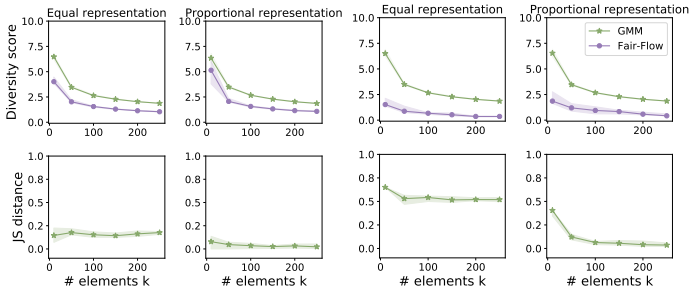
$$D_{KL}(p||z) = \sum_x p(x) \log\left(\frac{p(x)}{z(x)}\right)$$

In our evaluation, we use two real-world datasets and experiment with both binary and non-binary sensitive attributes. Below, we describe the different scenarios we explore for the two datasets.

Adult [23]. This is one of the most-commonly-used datasets in the fairness literature. It is extracted from the 1994 Census database and records income-related information. We preprocess the data to remove records with missing attributes, resulting in a data set with $n = 30,162$ tuples. The distance between two records is defined using the ℓ_2-norm over the six available numerical attributes (age, capital-gain, capital-loss, etc.) that we normalized to zero mean and unit variance.

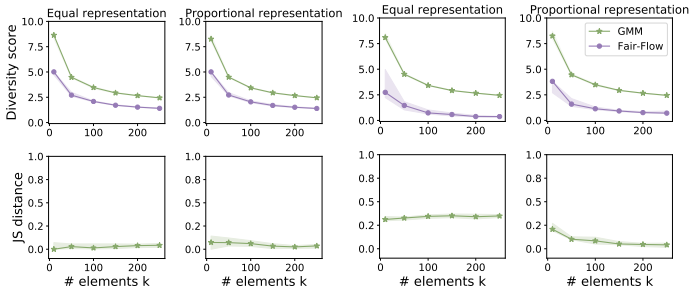
For the experiments on the Adult data, we first select sex as the sensitive attribute over which we specify fairness constraints so as to *proportionally* or *equally* represent the different demographic groups and range k from 10 to 250. Sex defines 2 demographic groups ($m = 2$): Female (9,782 records) and Male (20,380 records). We then select race as the sensitive attribute over which we specify fairness constraints. Race defines 5 demographic groups ($m = 5$): White (25,933 records), Asian-Pac-Islander (895 records), Amer-Indian-Eskimo (286 records), Black (2,817 records), and Other (231 records). In Figures 12(a), 12(b), we report the median value of the corresponding evaluation metric along with 90% confidence interval over 10 iterations of the experiment.

Bank Marketing [23, 40]. This dataset is related with direct marketing campaigns of a Portuguese banking institution and consists of $n = 41,188$ tuples. We define the distance between two records using the ℓ_2-norm over the ten numerical attributes (age, duration, etc.) in the Bank Marketing dataset that we normalized to zero mean and unit variance.



(a) Adult ($m = 2$) (b) Adult ($m = 5$)

Figure 12: Comparison of the Fair-Flow and the fairness-agnostic GMM in the Adult data.



(a) Bank ($m = 2$) (b) Bank ($m = 4$)

Figure 13: Comparison of the Fair-Flow and the fairness-agnostic GMM in the Bank Marketing data.

For the experiments on the Bank Marketing data, we select marital status as the sensitive attribute for the experiments over which we specify fairness constraints so as to *proportionally* or *equally* represent the different demographic groups and range k from 10 to 250. Marital status defines 4 demographic groups ($m = 4$): Married (24,928 records), Divorced (4,612 records), Single (11,568 records) and Unknown (80 records). Subsequently, we rearrange the partitions to define two groups; Married (24,928 records) and Non-married (16,260 records) and we experiment in a binary setting. In Figure 13(a), 13(b), we report the median value of the corresponding evaluation metric along with 90% confidence interval over 10 iterations of the experiment.

We note that, as expected, the diversity score drops when we incorporate fairness constraints. However, the price of fairness is small especially in the case of binary attribute settings. Nonetheless, we observe that the fairness-agnostic GMM can violate the constraints by a wide margin especially in the presence of high imbalanced groups and equal representation scenarios. (e.g in Figures 12(b), 13(b)). Overall, we believe that the *price of fairness* is a characteristic of the data (e.g how close the optimal solutions for the unconstrained and fair Max-Min problems respectively are) and algorithmic dependent (e.g how well an algorithm approximates

the optimal solution for Max-Min and fair Max-Min). In particular, notice that although our algorithms have a linear dependence on m, the price of fairness does not drastically increases as m grows.

B RESULTS ON FAIR k-CENTER CLUSTERING

In this paper, our primary focus has been on fair diversification using the Max-Min objective. In Section 1, we discussed how the unconstrained Max-Min diversification and the $k-$ center clustering problems are closely-related; notably, the best approximation algorithms for the unconstrained variants of both problems are essentially equivalent and result in the same approximation bound. In this section, we formally define the $k-$ center clustering, introduce its fair variant and discuss the known approximation results for this problem. 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.

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:

$$\underset{\mathcal{S}\subseteq\mathcal{U},|\mathcal{S}|=k}{\text{minimize}} \quad \max_{u\in\mathcal{U}} d(u,\mathcal{S})$$

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 [28] 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 [32]: 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. [32] provide a linear-time 5-approximation algorithm for the case of two classes ($m=2$), and a linear-time $\left(3\cdot 2^{m-1}-1\right)$-approximation for the general case, a result recently improved to $3(1+\epsilon)$ by Chiplunkar et al. [**?**].

Our contributions to this problem. In Section B.1, we provide a linear-time 3-approximation for fair k-center clustering, offering a slight improvement over the very recent result by Chiplunkar et al. [**?**] (noting that the two results were derived independently).

B.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. We further discuss how the work by Chiplunkar et al. [**?**] is related to ours and show which is the key idea that let us improve upon their method.

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 Algorithm 5 (Fair-Flow-Clust) 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 $d(y_i,y_j)>2\gamma$.

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 9. *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 4.2): 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,

Algorithm 5 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 $i \in [m]$

procedure FAIR-FLOW-CLUST
  $Y = \{y_1, \ldots, y_{k+1}\} \leftarrow \text{GMM}(\mathcal{U}, \emptyset, k+1)$
  for $j \in [k]$ **do**
    $D_j \leftarrow \{\text{argmin}_{x \in \mathcal{U}_i} d(x, y_j) : i \in [m]\}$
  if $d(y_{k+1}, \{y_1, \ldots, y_k\}) > 2\gamma$ **then return** $\emptyset$ ▷Abort
  else
    $Y = \{y_1, \ldots, y_t\}$ with minimum $t \leq k$ such that
      $d(y_{t+1}, \{y_1, \ldots, y_t\}) \leq 2\gamma$
  for $j \in [t]$ **do**
    $C_j \leftarrow \{x \in D_j : d(x, y_j) \leq \gamma\}$
  ▷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 $< t$ **then return** $\emptyset$ ▷Abort
  else ▷max flow is t
    $\forall (u_i, v_j)$ with flow add a node in C_j with color i to $\mathcal{S}$
  return $\mathcal{S}$

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 10. *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 [36, 37]. The total running time is therefore $O(kn + m^2k^2 \log k + mk^2 \log k)$ as required. □

Prior work has proposed another 3-approximation algorithm for fair k-center clustering [17], but the running time of that algorithm is quadratic with respect to the data size (n). Our 3-approximation bound for FAIR-FLOW-CLUST offers a slight improvement over the recent linear-algorithm with a $3(1+\epsilon)-$ approximation guarantee by Chiplunkar et al. [**?**].

Same as Chiplunkar et al. [**?**], we use the idea of guessing the optimum fair clustering radius using binary search and show how constructing a fair solution reduces to network flow; Chiplunkar et al. [**?**] first express the selection of a set of fair clusters as a matching problem (which they reduce to network flow). The key difference is that their approach generates the guesses using the continuous range method, or geometric guessing as they refer to it, which we also describe here. As a result both their running time and approximation has a dependence on ϵ. We show that discrete guessing over $O(k^2m^2)$ values, which is a smaller set than all the $O(n^2)$ pairwise distances of the elements in $\mathcal{U}$, provides a 3-approximation guarantee.

Algorithm 6 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
$k_1, k_2 \in \mathbb{Z}^+$
Output: at least k_i points in $\mathcal{U}_i$ for $i \in \{1, 2\}$
1: **procedure** FAIR$^+$-SWAP
  ▷Select as many bi-colored points as possible:
2: $\mathcal{S}_{\{1,2\}} \leftarrow$ maximal subset of $X_{\{1,2\}}$ with all points $\geq \gamma/4$ apart
  ▷Remove all points (of any color) arbitrarily close to $\mathcal{S}_{\{1,2\}}$:
3: $\mathcal{S}^- \leftarrow$ all the points in $\mathcal{U}$ that $< \gamma/4$ apart from a point in $\mathcal{S}_{\{1,2\}}$
4: $\mathcal{S}^+ \leftarrow \mathcal{U} \setminus \mathcal{S}^-$ ▷$\mathcal{S}^+ = \mathcal{S}^+_{\{1\}} \cup \mathcal{S}^+_{\{2\}} \subseteq X_{\{1\}} \cup X_{\{2\}}$
  ▷Select the missing points to satisfy the constraints:
5: Set $t = |\mathcal{S}_{\{1,2\}}|$
6: **if** $|\mathcal{S}^+ \cap \mathcal{U}_i| \geq k_i - t$ for $i \in \{1, 2\}$ **then**
7: $\mathcal{S}_{\{1\}} \cup \mathcal{S}_{\{2\}} \leftarrow$FAIR-SWAP$(\mathcal{S}^+, k_1 - t, k_2 - t)$
8: $\mathcal{S} \leftarrow \mathcal{S}_{\{1\}} \cup \mathcal{S}_{\{2\}} \cup \mathcal{S}_{\{1,2\}}$
9: **else**
10: $\mathcal{S} \leftarrow \emptyset$ ▷Abort
  return $\mathcal{S}$

C ADDITIONAL ALGORITHMS AND PROOFS

In this section, we provide proofs for the theoretical results described in Section 5. We further provide the pseudocode for the FAIR$^+$-SWAP algorithm (Algorithm 6), described in Section 5.1, and the pseudocode for the FAIR$^+$-FLOW algorithm (Algorithm 7), described in Section 5.2.

PROOF OF THEOREM 7. Let $O = O_{\{1\}} \cup O_{\{2\}} \cup O_{\{1,2\}}$ be the set that maximizes diversity and satisfies the fairness constraints with $\text{div}(O) = \ell^*_{\text{fair}}$.; thus, for any pair of optimal elements o_1, o_2 we have $d(o_1, o_2) \geq \ell^*_{\text{fair}}$. We prove that for any guess $\gamma \leq \ell^*_{\text{fair}}$, Algorithm 6 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, notice that $\text{div}(\mathcal{S}_{\{1,2\}}) \geq \gamma/4$ (by definition) and that $\mathcal{S}^-$ set in line 3 of algorithm 6 contains all the points in $\mathcal{S}_{\{1,2\}}$ and any points (of any color) arbitrarily close to them (including the points in $\mathcal{S}_{\{1,2\}}$). As a result, it follows that for any pair of points $x \in \mathcal{S}_{\{1,2\}}$ and $y \in \mathcal{S}^+$, we have $d(x, y) \geq \gamma/4$. Consequently, $\mathcal{S}^+$ consists of only single-colored points (if there were any bi-colored elements in $\mathcal{S}^+$ to satisfy the distance property, they would have been selected in the first step of the algorithm). Note that $\mathcal{S}^+ = \mathcal{S}^+_{\{1\}} \cup \mathcal{S}^+_{\{2\}}$ with $\mathcal{S}^+_{\{i\}} \subseteq X_{\{i\}}$ for $i \in \{1, 2\}$. 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 find a set of points to satisfy the fairness constraints that are at least $\gamma/4$ apart from each other.

Let $c^-_{\{1\}}, c^-_{\{2\}}, c^-_{\{1,2\}}$ be the number of optimal points in $O_{\{1\}}, O_{\{2\}}$ and $O_{\{1,2\}}$ respectively, present in $\mathcal{S}^-$ (noting that these are points that cannot be selected by FAIR-SWAP). Observe that

$$c^-_{\{1\}} + c^-_{\{2\}} + c^-_{\{1,2\}} \leq t$$

since at most one optimal point can be $\leq \gamma/4$ from a point in $\mathcal{S}_{\{1,2\}}$. For the sake of completeness, suppose that there exist a pair of o_1, o_2 optimal points such that $d(o_1, x) \leq \gamma/4$ and $d(o_2, x) \leq \gamma/4$ from

Algorithm 7 Fair$^+$-Flow: Overlapping classes

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
$k_1, \ldots, k_m \in \mathbb{Z}^+$
Output: at least k_i points in $\mathcal{U}_i$ for $i \in [m]$

procedure Fair$^+$-Flow
  Define
$$d_1 \leftarrow \frac{M\gamma}{3M-1} \text{ and } d_2 \leftarrow \frac{\gamma}{3M-1}$$
  $Z_{[m]} \leftarrow$ 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 such that all points in Z_L are $\geq d_1$ from every other element in
$$Z_L \cup \bigcup_{L' \in [m]: |L'| \geq t+1, L \subset L'} Z_{L'}$$
  Construct undirected graph G_Z 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
$$V = \{a, v_1, \ldots, v_t, b\} \cup \bigcup_{L \subset [m]: |L| > 0} \{u_L\}$$
$$E = \{(a, u_L) \text{ with capacity } c_L : \text{non-empty } L \subset [m]\} \cup \{(v_j, b) \text{ with capacity } 1 : j \in [t]\} \cup \{(u_L, v_j) \text{ with capacity } 1 : |Z_L \cap C_j| \geq 1\}$$
  Compute max a-b flow.
  if flow size $< \sum_{L \subset [m]} c_S$ **then return** $\emptyset$ ▷Abort
  else
    $\forall (u_L, v_j)$ with flow add a node in $C_j \in (\cap_{i \in S} \mathcal{U}_i)$ to $\mathcal{S}$
  return $\mathcal{S}$

some point $x \in \mathcal{S}_{\{1,2\}}$; by triangle inequality we get $d(o_1, o_2) \leq \gamma/2 \leq \ell^*_{\text{fair}}/2$ (contradiction).

As a result, it follows that there are 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 $O_{\{2\}}$ present in $\mathcal{S}^+$ that are $\geq \gamma \geq \ell^*_{\text{fair}}$ 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 $\text{div}(S) \geq \gamma/4$.

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. □

Proof of Lemma 3. 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 2, 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 2. 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 8. 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 3, this implies that all points in $f(O)$ are in different connected components. This implies that there exist connected components with the necessary requirements. □