Fair Diversification

Ashley Gao

January 2, 2024

Contents

1 Introduction

In the contemporary era, our lives are deeply intertwined with data, permeating diverse sectors such as machine learning, commerce, data mining, and healthcare. As datasets expand exponentially, the challenge to effectively utilize them becomes increasingly complex for both humans and computers. This necessitates the sampling of smaller, manageable datasets for practical applications. A critical aspect of this process is to ensure that these samples not only encapsulate the diversity inherent in the larger datasets but also guarantee the representation of each group. Moumoulidou et al. [2020] offers a tangible illustration of this approach in a real-world context. For instance, when searching for Nobel laureates online, the ideal outcome would be a diverse range of individuals in terms of age (reflecting diversity), while maintaining a balanced representation of genders (ensuring fairness). Another practical application of our research is depicted in Addanki et al. [2022], where they describe a scenario in selecting a restaurant in Manhattan, New York City. The objective is to suggest options that not only vary in location (diversity) but also offer a wide spectrum of cuisines (fairness). These examples underscore the relevance and applicability of our research in addressing the requirements of modern data utilization, and balancing diversity with fairness.

Paper Organization

This paper gives an introduction to the problem definition and key concepts in Section 2, followed by a review of the existing literature in the Fair Max-Min Diversification field in Section 3.

Our contributions are detailed in subsequent sections. Section 4 presents our extension of the original 2-approximation algorithm from a general metric space to an extended metric space, achieving a new approximation factor of $\frac{4}{c^2}$. Furthermore, in Section 5, we apply similar extensions to the 5-approximation models within the extended metric space. In Section 6, we not only refine the constant ϵ, but also elevate the fairness measure from $1-\epsilon$ to perfect fairness. Lastly, Section 7 discusses our attempts to generalize the FairSwap algorithm for $m \geq 3$, outlining the challenges encountered.

2 Definitions

For the Max-Min Diversification problem, there are 3 different modifications, leading to 2^3 versions that could be discussed. Currently, we do not consider more than 1 modifications applied to this problem, but they would lead to interesting open questions that could be discussed in the future. The problems of interest are namely:

1. Vanilla Fair Max-Min Diversification Problem
2. Fair Max-Min Diversification Problem with Extended Fairness Constraints
3. Fair Max-Min Diversification Problem with Overlapping Groups
4. Fair Max-Min Diversification Problem with Extended Metric Space

Instead of the Vanilla Fair Max-Min Diversification Problems that have already been stated in Addanki et al. [2022] and Moumoulidou et al. [2020] with metric space, I would also give the reader three more slightly different versions of the definition, which relaxes the fairness constraints, removes non-overlapping group restriction, or adopts the extended metric space as its problem domain. Problems 1 and 3 are mainly studied in Moumoulidou et al. [2020], and problem 2 is discussed in Addanki et al. [2022]. To the best of our knowledge, the last problem is first mentioned in this paper, and we give results for this problem in Section 4 and Section 5.

2.1 Approximation Ratio

The approximation ratio is a fundamental concept in algorithm design, particularly relevant to approximation algorithms. In this framework, let us denote the output of the approximation algorithm as ALG and the optimal solution as OPT. Typically, the approximation ratio is defined as $\alpha = ALG/OPT$. However, in the context of this paper, we focus on the max-min diversification problem, a maximization problem where OPT is always greater than or equal to ALG. To align better with the intuitive understanding obtained from other problems, we propose a modified notation for the approximation ratio: $\alpha = OPT/ALG$. This adjustment ensures that the ratio is always greater than or equal to 1, providing a more standardized metric for comparison and analysis. This is more intuitive for maximization problems because a ratio greater than 1 indicates how close the approximation (ALG) is to the optimal (OPT) and a ratio of 1 would mean the solution is exactly optimal.

Definition 1 (Approximation Ratio)**.** *Let OPT represent the optimal solution's value and ALG denote the value achieved by the approximation algorithm. The* ***approximation ratio*** *α for Max-Min Diversification Problem is*

$$\alpha = OPT/ALG$$

2.2 Vanilla Fair Max-Min Diversification Problem

Definition 2 (Metric Space)**.** *Let $\mathcal{X}$ be a universe of n elements and $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0^+$ be a distance function. $(\mathcal{X}, d)$ is a **metric space** if $\forall u, v, w \in \mathcal{X}$ satisfies:*

1. *(identity) $d(u, v) = 0 \iff u = v$*
2. *(symmetry) $d(u, v) = d(v, u)$*
3. *(positive) $d(u, v) \geq 0$*
4. *(triangle inequality) $d(u, w) \leq d(u, v) + d(v, w)$*

Definition 3 (Diversity)**.** *Throughout this paper, we refer to the **diversity** of a set S as*

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

In the context of this paper, diversity is defined as the minimum distance between two points in a given set.

Definition 4 (Vanilla Fair Max-Min Diversification Problem)**.** *Let $(\mathcal{X}, d)$ be a metric space where $\mathcal{X} = \bigcup_{i=1}^{m} \mathcal{X}_i$ is a universe of n elements positioned into m non-overlapping groups and $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0^+$ is a metric distance function.*

Further, let $k_1, k_2, \cdots, k_m$ be non-negatice integers with $k_i \leq |\mathcal{X}_i|$, $\forall i \in [m]$, and $k = \sum_{i=1}^{m} k_i$.

The problem of vanilla fair Max-Min diversification is defined as follows:

$$\max_{S \subseteq \mathcal{X}} \min_{u,v \in S, u \neq v} d(u, v)$$

or

$$\max_{S \subseteq \mathcal{X}} div(S)$$

subject to $|S \cap \mathcal{X}_i| = k_i$, $\forall i \in [m]$ (fairness constraints)

The objective of this problem is to find k_i points from each $\mathcal{X}_i$ to form a set S^*, such that all the possible selections of the set S satisfies the fairness constraints, $div(S^*) \geq div(S)$.

2.3 α-Approximation and β-Fairness

Let S^* be the optimal output for $\max_{S \subseteq \mathcal{X}} div(S)$, and $l^* = div(S^*)$, then in the context of our paper, we define α-approximation and β-fairness as follows:

A subset of points S is an α-approximation if $div(S) \geq l^*/\alpha$ and achieves β-fairness if $|S \cap \mathcal{X}_i| \geq \beta k_i$ for all $i \in [m]$, where $\alpha \geq 1$ and $\beta \in (0, 1]$. When $\beta = 1$, we call it achieves perfect fairness.

2.4 Fair Max-Min Diversification Problem with Extended Fairness Constraints

Using the β-fairness definition from Section 2.3, we can define the fair Max-Min diversification problem with **extended fairness constraints** as follows.

Definition 5 (Fair Max-Min Diversification Problem with Extended Fairness Constraints)**.** *Let $(\mathcal{X}, d)$ be a metric space where $\mathcal{X} = \bigcup_{i=1}^{m} \mathcal{X}_i$ is a universe of n elements positioned into m non-overlapping groups and $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0^+$ is a metric distance function.*

Further, let $k_1, k_2, \cdots, k_m$ be non-negatice integers with $k_i \leq |\mathcal{X}_i|, \ \forall i \in [m]$, and $k = \sum_{i=1}^{m} k_i$.

The problem is defined as follows:

$$\max_{S \subseteq \mathcal{X}} \ \min_{u,v \in S, u \neq v} d(u, v)$$

or

$$\max_{S \subseteq \mathcal{X}} \ div(S)$$

*subject to $|S \cap \mathcal{X}_i| \geq \beta k_i, \ \forall i \in [m]$ (**extended fairness constraints**)*

2.5 Fair Max-Min Diversification Problem with Overlapping Groups

Definition 6 (Fair Max-Min Diversification Problem with Overlapping Groups)**.** *Let $(\mathcal{X}, d)$ be a metric space where $\mathcal{X} = \bigcup_{i=1}^{m} \mathcal{X}_i$ is a universe of n elements positioned into m **overlapping** groups and $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0^+$ is a metric distance function.*

Further, let $k_1, k_2, \cdots, k_m$ be non-negatice integers with $k_i \leq |\mathcal{X}_i|, \ \forall i \in [m]$, and $k = \sum_{i=1}^{m} k_i$.

The problem is defined as follows:

$$\max_{S \subseteq \mathcal{X}} \ \min_{u,v \in S, u \neq v} d(u, v)$$

or

$$\max_{S \subseteq \mathcal{X}} \ div(S)$$

subject to $|S \cap \mathcal{X}_i| = k_i, \ \forall i \in [m]$ (fairness constraints)

2.6 Fair Max-Min Diversification Problem with Extended Metric Space

To define this problem, we first introduce the definition of **Extended Metric Space**.

Definition 7 (Extended Metric Space)**.** *$(\mathcal{X}, d)$ is an extended metric space with a factor $c \in (0, 2]$ if $\forall u, v, w \in \mathcal{X}$, the followings are satisfied:*

1. *(identity) $d(u, v) = 0 \iff u = v$*

2. *(symmetry)* $d(u, v) = d(v, u)$

3. *(positive)* $d(u, v) \geq 0$

4. *(triangle)* $c \cdot d(x, z) \leq d(x, y) + d(y, z)$

For an extended metric space, it is necessary to limit $c \in (0, 2]$ instead of all positive reals. The reason for this selection range is proved and justified in Section 9

With this in hand, we can formally define the problem:

Definition 8 (Fair Max-Min Diversification Problem with Extended Metric Space)**.** *Let* $(\mathcal{X}, d)$ *be an* ***extended*** *metric space where* $\mathcal{X} = \bigcup_{i=1}^{m} \mathcal{X}_i$ *is a universe of* n *elements positioned into* m *non-overlapping groups and* $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0^+$ *is a metric distance function.*

Further, let $k_1, k_2, \cdots, k_m$ *be non-negatice integers with* $k_i \leq |\mathcal{X}_i|$, $\forall i \in [m]$, *and* $k = \sum_{i=1}^{m} k_i$.

The problem is defined as follows:

$$\max_{S \subseteq \mathcal{X}} \min_{u,v \in S, u \neq v} d(u, v)$$

or

$$\max_{S \subseteq \mathcal{X}} div(S)$$

subject to $|S \cap \mathcal{X}_i| = k_i$, $\forall i \in [m]$ *(fairness constraints)*

3 Previous Work

In this section, we review prior research, categorizing studies based on whether they address the Max-Min Diversification problem within General Metric Spaces or Euclidean Metric Spaces. Within the context of General Metric Spaces, we explore two distinct scenarios: one characterized by the perfect fairness criterion, and the other by a more flexible, relaxed fairness standard.

3.1 General Metric Space Result With Perfect Fairness

3.1.1 4-Approximation, Perfect Fairness, $m = 2$

This section introduces the FairSwap algorithm, a 4-approximation method for the Vanilla Fair Max-Min Diversification problem (Problem 2.2) with perfect fairness, focusing on groups of size two, with time complexity $O(kn)$. For the Fair Max-Min Diversification Problem with Overlapping Groups (Problem 2.5), Moumoulidou et al. [2020] also gives a similar algorithm called Fair$^+$Swap to solve it with an approximation ratio of 4.

FairSwap is fundamentally a greedy algorithm. It uses the GMM algorithm, as mentioned in Ravi et al. [1994] (not to be confused with Gaussian Mixture Modelling), to find an initial solution by treating all groups as the same. Then, it adjusts this solution for fairness by modifying groups as necessary. The pseudo-code to this algorithm is presented in Figures 1 and 2. For the FairSwap Algorithm, proofs of its approximation ratio and running time are detailed in Moumoulidou et al. [2020], which we invite interested readers to read. Our paper also extends the algorithm to extended metric spaces, with proofs in Section 4. Section 7 discusses whether it is possible to extend this approach to $m \geq 3$ cases, highlighting challenges with the optimal sub-structure property.

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**
    $x \leftarrow \underset{u \in \mathcal{U}}{\operatorname{argmax}} \min_{s \in \mathcal{S} \cup I} d(u, s)$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{x\}$
return $\mathcal{S}$

Figure 1: GMM Algorithm

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 $i \in \{1, 2\}$

procedure FAIR-SWAP
▷Color-Blind Phase:
  $\mathcal{S} \leftarrow \text{GMM}(\mathcal{U}, \emptyset, k)$
  $\mathcal{S}_i = \mathcal{S} \cap \mathcal{U}_i$ for $i \in \{1, 2\}$
▷Balancing Phase:
  Set $U = \text{argmin}_i(k_i - |\mathcal{S}_i|)$ ▷Under-satisfied set
  $O = 3 - U$ ▷Over-satisfied set
  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 $(\mathcal{S}_U \cup E) \cup (\mathcal{S}_O \setminus R)$

Figure 2: Fair Swap

The Fair$^+$Swap algorithm is a variant of the FairSwap Algorithm, distinguished primarily by its input requirements. Unlike FairSwap, which necessitates disjoint group inputs, Fair$^+$Swap is designed to handle cases with overlapping groups (Problem 2.5). Moumoulidou et al. [2020] addresses this overlap by initially selecting points that ensure a minimum separation of $\gamma/4$. Subsequently, FairSwap is applied to the remaining universe, excluding the overlapping segments of the groups. Figure 3 shows the pseudo-code of the algorithm, which helps readers to understand how it works. Fair$^+$Swap shares the same approximation ratio as the FairSwap, which is 4.

3.1.2 5-Approximation, Perfect Fairness, $m = O(1), k = o(\log n)$

Moumoulidou et al. [2020] states a 5-approximation algorithm with perfect fairness, which is practical for small k and m (Figure 4). The running time of this algorithm is $O(kn + k^2(em)^k)$. Furthermore, it would be linear in n if $m = O(1), k = o(\log n)$ is satisfied.

This algorithm uses the GMM algorithm as a building block as well. The basic idea of this algorithm is to select k points from each group at first, which means that at most, we have km points that are selected. Later on, we find our final solution through an exhaustive search. This is the main reason that we want to keep k and m small since the running time would depend exponentially on k.

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

Input: $\mathcal{U}_1, \mathcal{U}_2$: Universe of available elements
$\gamma \in \mathbb{R}$: A guess on the optimum fair diversity
$k_1, k_2 \in \mathbb{Z}^+$

Output: at least k_i points in $\mathcal{U}_i$ for $i \in \{1, 2\}$

procedure FAIR$^+$-SWAP
  $\mathcal{S}_{\{1,2\}} \leftarrow$ maximal subset of $X_{\{1,2\}}$ with all points $\geq \gamma/4$ apart
  $\mathcal{S}^- \leftarrow$ all the points in $\mathcal{U}$ that $< \gamma/4$ apart from a point in $\mathcal{S}_{\{1,2\}}$
  $\mathcal{S}^+ \leftarrow \mathcal{U} \setminus \mathcal{S}^-$ ▷$\mathcal{S}^+ = \mathcal{S}^+_{\{1\}} \cup \mathcal{S}^+_{\{2\}} \subseteq X_{\{1\}} \cup X_{\{2\}}$

▷Select the missing points to satisfy the constraints:
  Set $t = |\mathcal{S}_{\{1,2\}}|$
  if $|\mathcal{S}^+ \cap \mathcal{U}_i| \geq k_i - t$ for $i \in \{1, 2\}$ **then**
    $\mathcal{S}_{\{1\}} \cup \mathcal{S}_{\{2\}} \leftarrow$FAIR-SWAP$(\mathcal{S}^+, k_1 - t, k_2 - t)$
    $\mathcal{S} \leftarrow \mathcal{S}_{\{1\}} \cup \mathcal{S}_{\{2\}} \cup \mathcal{S}_{\{1,2\}}$
  else
    $\mathcal{S} \leftarrow \emptyset$ ▷Abort
return $\mathcal{S}$

Figure 3: Fair$^+$ Swap

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 $i \in [m]$

procedure FAIR-GMM
  for $i \in [m]$ **do** $Y_i \leftarrow \text{GMM}(\mathcal{U}_i, \emptyset, k)$
  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.

Figure 4: Fair GMM

The basic idea for proving the approximation ratio is, for the k points that are returned by any group $\mathcal{U}_i$ in our first step of the algorithm, as known as Y_i, Y_i will remove at most $k - k_i$ points because it is too close from other colors points, and the remaining k_i points will still maintain a high diversity meanwhile satisfy the fairness constraints.

In our paper, we show that this algorithm could extend to a broader metric space, achieving a 5-approximation. Detailed explanations and demonstrations of this extension are presented in Section 5.

3.1.3 $m + 1$-approximation, Perfect fairness

The algorithm introduced in this section employs a network flow approach and offers an approximation factor of $m + 1$, ensuring perfect fairness. Addanki et al. [2022] proposed this algorithm to address Problem 2.2. The core concept involves iteratively creating multiple clusters. These clusters are formed by grouping points that are within a distance less than $\frac{\gamma}{m+1}$, where γ represents an estimated optimal value l^*. Subsequently,

a max-flow operation is executed to select a sufficient number of clusters. This selection adheres to fairness constraints, as it allows for the choice of representative points from each cluster. The algorithm's steps are outlined in Figure 6.

In terms of constructing the flow graph, we use the following procedures. Consider groups denoted as $u_1, u_2, \ldots, u_m$ and clusters represented as $v_1, v_2, \ldots$. We designate a as the source and b as the sink. Initially, we establish connections from the source a to each group u_i, assigning a weight k_i to each edge, where $i \in [m]$. Subsequently, we create edges between each corresponding pair of u_i and v_j with a weight of 1. Similarly, we connect each cluster v_j to the sink b with edges also weighted at 1. We give a concrete example in Figure 5.

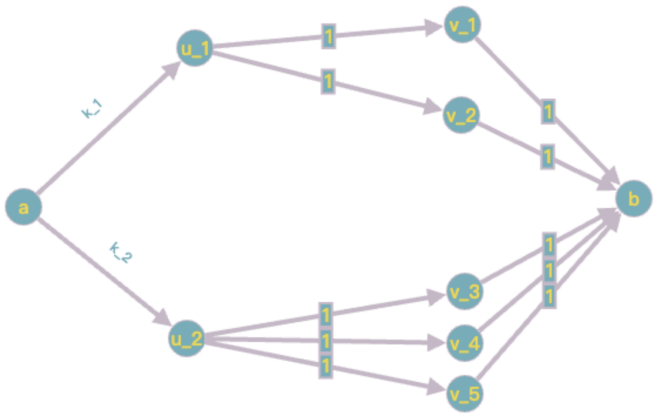

Figure 5: Network Flow Example

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

This theorem is proved in Addanki et al. [2022]. It is worth mentioning that the preceding version of this algorithm, as proposed by Moumoulidou et al. [2020], employs a network flow approach and presents a $3m-1$ approximation. In the same work, Moumoulidou et al. [2020] extends the application scope of this approximation algorithm from disjoint to overlapping groups, where points in the metric space can belong to multiple groups. This extension results in an outcome of $3\begin{pmatrix} m \\ |m/2| \end{pmatrix} - 1$ for overlapping groups. We hypothesize that by applying the novel $m+1$ algorithm, this result could potentially be enhanced to $\begin{pmatrix} m \\ |m/2| \end{pmatrix} + 1$.

Algorithm 4 FAIR-GREEDY-FLOW

Input: $\mathcal{X} = \bigcup_{i=1}^{m} \mathcal{X}_i$: 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 $\mathcal{X}_i$ for $i \in [m]$.

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

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

Set $\mathcal{S} \leftarrow \emptyset$. Compute maximum a-b flow in G using Ford-Fulkerson algorithm [25].
if flow size $< k = \sum_i k_i$ **then return** $\emptyset$ ▷Abort
else ▷max flow is k
  $\forall (u_i, v_j)$ with flow equal to 1, add the point in D_j with group i to $\mathcal{S}$.
end if
return $\mathcal{S}$.

Figure 6: Fair Greedy Flow

3.2 General Metric Space Result With Relaxed Fairness

3.2.1 2-Approximation, Expected Fairness

In this section, we explore a 2-approximation algorithm designed for Max-Min Diversification with an emphasis on expected fairness from Addanki et al. [2022]. It is crucial to understand that this algorithm does not guarantee fairness per se, but instead provides an expectation that fairness constraints are likely to be met. The core approach of the algorithm is to apply a randomization technique and Integer Linear Programming (ILP) relaxation. To address this problem, we construct and solve a linear programming model and then convert the solution to an integer.

To construct the LP model, we first need to define a notation and some variables.

Definition 9 (Ball)**.** *A ball of radius r centered at $p \in \mathcal{X}$ as the set of all points q in $\mathcal{X}$ such that $d(p, q) < r$, that is $\mathbf{B}(p, r) = \{q \in \mathcal{X} \mid d(p, q) < r\}$*

In accordance with the definition presented in Problem 2.2, we introduce variables $x_1, \ldots, x_n$, each ranging between 0 and 1, to serve as indicators. These indicators determine whether we include point p in the proposed solution set.

The LP model for this algorithm is defined as follows:

$$\sum_{p_j \in \mathcal{X}_i} x_j \geq k_i \quad \forall i \in [m].$$
$$\sum_{p_\ell \in \mathbf{B}(p,\gamma/2)} x_\ell \leq 1 \quad \forall p \in \mathcal{X}.$$
$$x_j \geq 0 \quad \forall j \in [n].$$

The first constraint is designed to ensure fairness for each group. The second constraint requires that the distance between all points in our chosen solution set is at least $\gamma/2$. For the third constraint, we ensure that the variable x_j has a value from 0 to 1 when combined with the second constraint, which meets our requirement stated before.

We then outline the algorithm in a step-by-step fashion.

Algorithm Outline - 2-Approximation with Expected Fairness

1. Parameter Selection: Choose a parameter γ such that it is maximized while ensuring $\gamma \leq l^*$.
2. Linear Programming (LP) Solution: Solve the aforementioned LP to obtain an initial solution.
3. From LP to Integer Linear Programming (ILP) Solution:
   - Let $x_1^*, \ldots, x_n^*$ be the solution of the previous LP
     $n' = |\{j : x_j^* > 0\}|$. i.e., n' is the number of the element with $x_j^* > 0$
   - By sampling without replacement, generate a sequence σ with length n'.
     - The procedure for generating the sequence:
       For each step t, let R_t represent the set $[n'] \setminus \{\sigma(1), ..., \sigma(t-1)\}$.
       Then we choose i-th element of σ following the probability distribution

$$P[\sigma(t) = j] = \frac{x_j^*}{\sum_{l \in R_t} x_l^*}$$

   - Subsequently, construct the solution set with a 2-approximation approach, ensuring expected fairness.
     - The procedure for generating the sequence:
       Including the point p_j in the solution set if and only if $\sigma(j) \leq \sigma(l)$ for all $p_l \in \mathbf{B}(p_j, \gamma/2)$.
       Formalized as:

$$p_j \in S \iff \forall p_l \in \mathbf{B}(p_j, \gamma/2), \ \ \sigma(j) \leq \sigma(l)$$

4. Solution Set: The final solution set S is derived, offering a 2-approximation with expected fairness.

The ability of the algorithm to achieve a 2-approximation is guaranteed via our linear programming (LP) framework. Regarding the expected fairness property, a detailed justification is provided in Addanki et al. [2022]. We intend to recapitulate the essentials of their proof here.

Proof. Let A_t to be the event that $(d(p_{\sigma(t)}, p_j) < \gamma/2) \wedge (d(p_{\sigma(t')}, p_j) \geq \gamma/2)$ for all $t' < t$.

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

Incorporating the first linear programming constraint, we have $\mathbb{E}[|S \cap \mathcal{X}_i|] = \sum_{p \in \mathcal{X}_i} \Pr[p \in S] \geq \sum_{p \in \mathcal{X}_i} x_p^* \geq k_i$. □

3.2.2 6-Approximation, $(1-\epsilon)$-Fairness in $(1-\delta)$ Probability

In this section, we present an algorithm that achieves a 6-approximation, ensuring fairness of $(1-\epsilon)$ (where $\epsilon \geq \sqrt{\frac{3\log(2m)}{k_i}}$) with the success rate $(1-\delta)$ (where δ could be arbitrarily small by re-run the algorithm $\log \delta^{-1}$ times). This algorithm addresses Problem 2.4. To obtain a sufficiently accurate fairness requirement, it is advantageous to have a larger input value of k with this algorithm. This algorithm, initially proposed in Addanki et al. [2022], develops from the earlier 2-Approximation with Expected Fairness algorithm discussed in Section 3.2.1. It modifies the radius of the ball in the previous linear programming constraint from 2 to 6 and introduces an additional non-linear constraint to achieve the $1-\epsilon$ fairness target. The solution to this Non-Linear Programming problem is derived by adapting the results from the previous linear programming approach. For this section, we aim to demonstrate the process of generating a solution set from modifying existing linear programming results, then show that it also meets the Non-Linear programming constraints, and subsequently verify the claimed approximation and fairness ratios.

The revised "Non-LP" model is

$$\begin{aligned}
\sum_{p_j \in \mathcal{X}_i} y_j \geq k_i \quad \forall i \in [m]. \\
\sum_{p_\ell \in \mathbf{B}(p, \gamma/6)} y_\ell \leq 1 \quad \forall p \in \mathcal{X}. \\
y_j \geq 0 \quad \forall j \in [n].
\end{aligned}$$

$$\forall \ell \in [m],\ \forall p_i, p_j \in \mathcal{X}_\ell,\ (0 < y_i) \wedge (0 < y_j) \implies d(p_i, p_j) \geq \frac{\gamma}{3}$$

We change the ball radius from 2 (Section 3.2.1) to 6 and add the last constraint, which is later used to justify the $1-\epsilon$ fairness bound. I would like to state the process of adopting the solution from Section 3.2.1 to this new Non-Linear Programming model.

Algorithm Outline: Construct solution set from Section 3.2.1

1. Generate $\{x_j^*\}_{j \in [n]}$ using the algorithm in Section 3.2.1, without ILP rounding process.

2. Generate $\{y_j^*\}_{j\in[n]}$ from $\{x_j^*\}_{j\in[n]}$

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

   - $y_j^* \leftarrow \left(\sum_{p_\ell \in \mathbf{B}(p_j, \frac{\gamma}{3}) \cap \mathcal{X}_i} x_\ell^*\right)$
   - $y_\ell^* \leftarrow 0$ for all $p_\ell \in \mathbf{B}\left(p_j, \frac{\gamma}{3}\right) \cap (\mathcal{X}_i \backslash \{p_j\})$

   For this step, we move the weights of the points that are in the same group as p_j and close to p_j to p_j

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

3. From LP to Integer Linear Programming (ILP) Solution (similar to Section 3.2.1):,

   - Let $n' = |\{j : y_j^* > 0\}|$. i.e., n' is the number of the element with $y_j^* > 0$
   - By sampling without replacement, generate a sequence σ with length n'.
     - The procedure for generating the sequence:
       For each step t, let R_t represent the set $[n'] \setminus \{\sigma(1), ..., \sigma(t-1)\}$.
       Then we choose i-th element of σ following the probability distribution

$$P[\sigma(t) = j] = \frac{y_j^*}{\sum_{l \in R_t} y_l^*}$$

   - Subsequently, construct the solution set with a 6-approximation approach, ensuring $(1-\epsilon)$ fairness.
     - The procedure for generating the sequence:
       Including the point p_j in the solution set if and only if $\sigma(j) \leq \sigma(l)$ for all $p_l \in \mathbf{B}(p_j, \gamma/6)$.
       Formalized as:

$$p_j \in S \iff \forall p_l \in \mathbf{B}(p_j, \gamma/6),\ \ \sigma(j) \leq \sigma(l)$$

4. Solution Set: The final solution set S is derived, offering a 2-approximation with expected fairness.

Next, we will demonstrate that the solution generated adheres to the established framework.

Remark 1. **Generated $\{y_j^*\}_{j\in[n]}$ is a valid solution of given "Non-LP"(i.e. 4 constraints)**

Proof. 1. The constraint $\sum_{p_j \in \mathcal{X}_i} y_j \geq k_i$ is fulfilled. This is evident from the 2-Approximation LP model, where $\sum_{p_j \in \mathcal{X}_i} x_j \geq k_i$. Our algorithm ensures that $\sum_{p_j \in \mathcal{X}_i} x_j$ equals $\sum_{p_j \in \mathcal{X}_i} y_j$, thereby satisfying this constraint.

2. The constraint $\sum_{p_\ell \in \mathbf{B}(p,\gamma/6)} y_\ell \leq 1$ holds true. In the 2-Approximation model, it is established that $\sum_{p_\ell \in \mathbf{B}(p,\gamma/2)} y_\ell \leq 1$. Consequently, we have:

$$\sum_{p_\ell \in \mathbf{B}(p,\gamma/6)} y_\ell \leq \sum_{p_\ell \in \mathbf{B}(p,\gamma/3+\gamma/6)} y_\ell \leq \sum_{p_\ell \in \mathbf{B}(p,\gamma/2)} y_\ell \leq 1$$

3. This constraint is met since $x_j \geq 0$. Our algorithm either redistributes y_j as a sum of adjacent values or assigns it a value of 0. Thus, $y_j \geq 0$.

4. By redistributing the values of all points within the same group and inside a ball with a radius of $\gamma/3$ to a single point, this constraint is inherently met.

In conclusion, the solution $\{y_j^*\}_{j\in[n]}$ is valid within the specified "Non-LP" constraints. □

Then, for the 6-approximation and $(1-\epsilon)$ fairness properties, we restate the following theorem from Addanki et al. [2022] and give a more detailed proof than the original text:

Theorem 2. *Let $\epsilon \geq \sqrt{\frac{3\log(2m)}{k_i}}$ for all $i \in [m]$, $\delta \in \mathbb{R}$. There is a poly(n, k, δ^{-1}) time algorithm that returns a subset of points with diversity $l^*/6$ and includes $(1-\epsilon)k_i$ points in each group $i \in [m]$ with probability at least $(1-\delta)$ if we run the algorithm $\log \delta^{-1}$ times.*

Notably, in the original text of Addanki et al. [2022], the theorem is presented differently. In my paper, I have changed the notation from "`Assume` $k_i \geq 3\epsilon^{-2}\log(2m)$" to "`Let` $\epsilon \geq \sqrt{\frac{3\log(2m)}{k_i}}$". This change shifts our focus from variable k to ϵ. This is important for Problem 2.4, because we cannot assume all k_i meet our criteria all the time. Instead, by modifying the problem, the algorithm becomes more flexible and can handle a variety of inputs. Though it still performs best with larger k values.

Proof. Let $Y_p = 1$ be the event if the point $p \in \mathcal{X}$ is included in the output S, let $i \in [m]$.
Since we have the Proof from Section 3.2.1 for the 2-approximation LP algorithm, similarly, it would also applied to the current ball with radius $\gamma/6$. Hence we have

$$\mathbb{E}[\sum_{p\in\mathcal{X}_i} Y_p] \geq k_i$$

For two disjoint balls $\mathbf{B}(p_i, \gamma/2)$ and $\mathbf{B}(p_j, \gamma/2)$, the relative ordering in σ of the element in these two balls, i.e. $\{l : p_l \in \mathbf{B}(p_i, \gamma/2)\}$ and $\{l : p_l \in \mathbf{B}(p_j, \gamma/2)\}$ are independent. Therefore, $\{Y_p\}_{p\in\mathcal{X}_i}$ is an independent variable. Then, we could apply Chernoff Bound, which is defined as the following:

Definition 10. ***Multiplicative Chernoff Bound:***
Suppose $X_1, \ldots, X_n$ are independent random variables taking values in $0, 1$. Let $X = X_1 + X_2 + \ldots + X_n$ denote their sum and with expected value $\mathbb{E}[X]$, the Chernoff bound for any $\delta > 0$ is given by:
- ***Upper Tail Bound:***

$$P(X \geq (1+\delta)\mathbb{E}[X]) \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mathbb{E}[X]}$$

- ***Lower Tail Bound:***

$$P(X \leq (1-\delta)\mathbb{E}[X]) \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mathbb{E}[X]}$$

Then we have

$$\Pr[\sum_{p\in\mathcal{X}_i} Y_p \leq (1-\epsilon)k_i] \leq \exp\left(-\epsilon^2 k_i/3\right) \leq \frac{1}{2m}$$

The detail of getting this inequality from Chernoff Lower Tail Bound is skipped for now since it is similar to an enhanced analysis in Section 6. This enhanced analysis improves the constant from $\exp\left(-\epsilon^2 k_i/3\right)$ to $\exp\left(-\epsilon^2 k_i/2\right)$, and also demonstrates that our bound is the best bound we can get from the Chernoff framework.

Using the application of union bound, each time we run the algorithm, it has a probability of at least $1/2$ such that $(1-\epsilon)$ fairness would be satisfied for all of m groups, i.e. $|S \cap \mathcal{X}_i| \geq (1-\epsilon)k_i$ for all $i \in [m]$. If we would like to increase the probability of having at least 1 trail to succeed, we can repeat the process $\log \delta^{-1}$ times to get a probability of $1-\delta$. □

In Section 3.1.2, we present a 5-approximation algorithm that ensures perfect fairness. The main difference between the 5-approximation algorithm and our current 6-approximation is not only the fairness; 5-approximation algorithm works best for smaller values of k, as they require a small value of k to have polynomial running time, i.e. $k = o(\log n)$. On the other hand, the 6-approximation algorithm is suited for cases with larger k values. Therefore, the choice of which algorithm to use should be based on the size of the input parameters $k_1, \ldots, k_m$.

In our paper, we present improvements to the algorithm discussed in this section. Specifically, we have refined the algorithm to achieve a more efficient constant, reducing ϵ from 3 to 2. Additionally, we improve fairness from $1-\epsilon$ to attain complete fairness. The details of these improvements are elaborated in Section 6.

3.3 Euclidean Metric Space

3.3.1 1D Euclidean Metric Space

In terms of the vanilla max-min problem (Problem 2.2) in 1D Euclidean space, Addanki et al. [2022] developed a dynamic programming (DP) algorithm capable of solving this problem. The proposed algorithm (Figure 7) exhibits a time complexity of $O(n^4 \prod_{i=1}^{m}(k_i+1))$. Introducing an additional constraint, specifically setting $m = 1$ so that all points are in the same group, Wang and Kuo [1988] introduced a more efficient DP algorithm (Figure 8) for this case with time complexity of $O(kn + n\log n)$.

3.3.2 Constant Euclidean Metric Space

In this section, we present two algorithms proposed by Addanki et al. [2022]. The first algorithm is applicable to any metric space characterized by a low doubling dimension, offering a $(1+\epsilon)$-approximation solution that ensures perfect fairness. In contrast, the second algorithm extends the requirements to include not only the low doubling dimension property but also a constant Euclidean metric space. This algorithm provides a $(1+\epsilon)$-approximation solution with a slightly reduced fairness of $(1-\epsilon)$. Notably, the second algorithm

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

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

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

Let $n \leftarrow |\bigcup_{i=1}^{m} \mathcal{X}_i|$ and initialize $H \in \{0,1\}^{(k_1+1)\times\ldots(k_m+1)\times n}$ to 0.
Set $H[0,\ldots,0,0] \leftarrow 1$, $H[0,\ldots,0,1] \leftarrow 1$, and if $p_1 \in \mathcal{X}_\ell$, $H[0,\ldots,\underbrace{1}_{\text{index } \ell},\ldots,0,1] \leftarrow 1$.
for $j = 2$ to n **do**
  Let $i \in [m]$ satisfy $p_j \in \mathcal{X}_i$.
  Let $j' = \max(\{0\} \cup \{j' \in [n] : p_{j'} + \gamma \leq p_j\})$.
  for $k_1' \in \{0,\ldots,k_1\}, \ldots, k_m' \in \{0,\ldots,k_m\}$ **do**
    $H[k_1', \cdots, k_m', j] \leftarrow H[k_1', \cdots, k_m', j-1]$.
    If $k_i' \geq 1$, $H[k_1', \cdots, k_m', j] \leftarrow H[k_1', \cdots, k_i'-1, \ldots, k_m', j'] \vee H[k_1', \cdots, k_m', j-1]$.
  end for
end for
return $H[k_1, k_2, \cdots, k_m, n]$.

Figure 7: DP Algorithm for $\mathbb{R}^1$

Algorithm B1

for $i = 2$ **to** n **do** $d_{i2} = p_i - p_1$ **endfor**
for $j = 3$ **to** p **do**
  $h(j,\ j) = j-1$
  $d_{jj} = \min\{d_{j-1,j-1},\ p_j - p_{j-1}\}$
  for $i = j+1$ **to** n **do**
    $h = h(i-1,\ j)$
    while $h < i-1$ **and**
      $\min\{d_{h,j-1},\ p_i - p_h\}$
        $\leqslant \min\{d_{h+1,j-1},\ p_i - p_{h+1}\}$
      do $h = h+1$
    endwhile
    $h(i,\ j) = h$
    $d_{ij} = \min\{d_{h,j-1},\ p_i - p_h\}$
  endfor
endfor

Figure 8: DP Algorithm for $\mathbb{R}^1$ and $m = 1$

demonstrates a faster computational performance, due to the polynomial dependence on n and k.

Addanki et al. [2022] proposed an algorithm designed for metric spaces characterized by low doubling dimensions. A prevalent example of such a metric space is the Euclidean Metric Space with a constant dimension. Let λ represent the doubling dimension of the space $\mathcal{X}$. Before proceeding, we will define key concepts necessary for understanding our approach.

Definition 11 (Doubling Dimension)**.** *Let $(\mathcal{X}, d)$ be a metric space. The doubling dimension of $\mathcal{X}$ is the smallest integer λ such that any ball $\mathbf{B}(p,r)$ of radius r around a point $p \in \mathcal{X}$ can be covered using at most $(r/r')^{\lambda}$ balls of radius r'. The Euclidean metric on $\mathbb{R}^D$ has doubling dimension $O(D)$*

Definition 12 (Coreset for Fair Max-Min)**.** *A set $\mathcal{T} \subseteq \mathcal{X}$ is an α-coreset if there exists a subset $\mathcal{T}' \subseteq \mathcal{T}$ with $|\mathcal{T}' \cap \mathcal{X}_i| = k_i, \forall i \in [m]$ and $\operatorname{div}(\mathcal{T}') \geq \ell^*/\alpha$*

Definition 13 (Composable coreset for Fair Max-Min)**.** *A function $c(\mathcal{X})$ that maps a set of elements to a subset of these elements computes an α-composable coreset for some $\alpha \geq 1$, if for any partitioning of $\mathcal{X} = \bigcup_j \mathcal{Y}_j$ and $\mathcal{T} = \bigcup_j c(\mathcal{Y}_j)$, there exists a set $\mathcal{T}' \subseteq \mathcal{T}$ with $|\mathcal{T}' \cap \mathcal{X}_i| = k_i, \forall i \in [m]$ such that $\operatorname{div}(\mathcal{T}') \geq \ell^*/\alpha$.*

The study presented in Addanki et al. [2022] outlines a method for creating a $(1+\epsilon)$ coreset $\mathcal{T}$, which includes a subset $\mathcal{T}'$ with $div(\mathcal{T}') \geq l^*/(1+\epsilon)$. For the details of this procedure, readers are encouraged to refer to the detailed explanation provided in Addanki et al. [2022].

Theorem 3. *There is an algorithm that returns a $(1+\epsilon)$-coreset of size $O\left((8/\epsilon)^{\lambda} km\right)$ in metrics of doubling dimension λ with a running time $O\left((8/\epsilon)^{\lambda} kmn\right)$.*

Based on Theorem 3, from the coreset $\mathcal{T}'$, we can traverse all the subsets of $\mathcal{T}'$ and return the set with the highest diversity and perfect fairness. The running time of the algorithm is $O(2^{O(k)} + nk)$, where m, λ are constants.

Since iterating all subsets of $\mathcal{T}$ can be time intensive, Addanki et al. [2022] introduces a new approach that uses the coreset $\mathcal{T}$ that is constructed previously, returning a $(1+\epsilon)$ approximation set with a trade-off in

$(1-\epsilon)$ fairness. This approach necessitates the use of a constant Euclidean space as the metric. Missing details can be found in Addanki et al. [2022].

Theorem 4. *If $\gamma \geq \ell^*/(1+\epsilon)$, the algorithm returns a set $\mathcal{S}$ such that $div(\mathcal{S}) \geq \ell^*/(1+\epsilon)$ and $|\mathcal{S} \cap \mathcal{X}_i| \geq k_i(1-\epsilon), \forall i \in [m]$ with probability at least $1-\delta$. For constant D, m, the running time is $O(nk+\text{poly}(1/\epsilon, k, \log(1/\delta)))$.*

4 Proof for $m = 2$ settings for extended metric space

Theorem 5. *FairSwap is a $\frac{4}{c^2}$-approximation algorithm for the fair diversification problem with extended metric factor c when $m = 2$ that runs in time $O(kn)$*

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**
    $x \leftarrow \underset{u \in \mathcal{U}}{\text{argmax}} \min_{s \in \mathcal{S} \cup I} d(u, s)$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{x\}$
  return $\mathcal{S}$

Figure 9: GMM Algorithm

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 $i \in \{1, 2\}$

procedure FAIR-SWAP
▷Color-Blind Phase:
  $\mathcal{S} \leftarrow \text{GMM}(\mathcal{U}, \emptyset, k)$
  $\mathcal{S}_i = \mathcal{S} \cap \mathcal{U}_i$ for $i \in \{1, 2\}$
▷Balancing Phase:
  Set $U = \text{argmin}_i(k_i - |\mathcal{S}_i|)$ ▷Under-satisfied set
  $O = 3 - U$ ▷Over-satisfied set
  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 $(\mathcal{S}_U \cup E) \cup (\mathcal{S}_O \setminus R)$

Figure 10: Fair Swap

This proof is inspired by Moumoulidou et al. [2020].

I simply extend the result regarding part of Moumoulidou et al. [2020] from the metric space to an extended metric space, which is Fair Max-Min Diversification with $m = 2$ (i.e., Section 3.2.1). We analyze the approximation ratio and time complexity of the problem in extended metric space, when applying Fair Swap Algorithm in Moumoulidou et al. [2020]. The results are: the time complexity remains $O(kn)$, and the approximation ratio increases to $\frac{4}{c^2}$, where c is constant regarding the property of the extended metric.

Fair Swap Algorithm is attached in Figure 10. This algorithm is presented by Moumoulidou et al. [2020], which uses the GMM algorithm (attached in Figure 9) that is proposed in Ravi et al. [1994] as a building block.

Define the problem domain. I would like to show the proof of the approximation ratio and time complexity for the problem **Fair Max-Min Diversification Problem with Extended Metric Space** with the given scenario:

- Two-group ($m = 2$)

- Non-overlapping
- $(\mathcal{X}, d)$ is the extended metric space with constant $c \in \mathbb{R}$
- Applying Fair Swap Algorithm (Algorithm 2)

Running Time Analysis. Since it is running the same algorithm as the non-extended version that has been proposed by Moumoulidou et al. [2020], the running time would follow the previous result, which is $O(kn)$

Approximation-Factor Analysis. Let's define the input and output.

- Input: A set of points $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$ and $k_1, k_2 \in \mathbb{R}^+$, with $k_i \leq |\mathcal{U}_i|, \forall i \in \{1, 2\}$
- Output: A subset of $\mathcal{U}$ with k_i points from each $\mathcal{U}_i$ partition such that the div(S) is maximized

Let's define the terms used in the proof.

- S^*: The set of k points in $\mathcal{U}$ that maximize the diversity when there are no fairness constraints, which means that we assume all the points are in the same group and disregard fairness.
- l^*: The divergence of the set S^*, which is $l^* = div(S^*)$
- $\mathcal{F}^* = \mathcal{F}_1^* \cup \mathcal{F}_2^*$: $\mathcal{F}^*$ is the optimal solution for the Fair Max-Min diversification, and

$$\mathcal{F}_1^* = \mathcal{F}^* \cap \mathcal{U}_1, \qquad \mathcal{F}_2^* = \mathcal{F}^* \cap \mathcal{U}_2$$

- l^*_{fair}: The divergence of the set $\mathcal{F}^*$, which is $l^*_{\text{fair}} = div(\mathcal{F}^*)$
- $S, S_1, S_2, S_U, S_O, E, R, U, O$: Adopt the same meaning defined in the Algorithm.
  - S is the intermediate output of the algorithm with no fairness constraints (output for the Color-Blind Phase).
  - S_1 is the points that are both in S and group 1, while S_2 is the points that are both in S and group 2.
  - S_U is one of the S_1 or S_2 which under-satisfy the constraint for the group (which means that we need to add more points from S_U to satisfy the fairness constraint), while S_O is the other set which over-satisfied.
  - E is the intermediate output of the GMM algorithm. This algorithm greedily chooses the points that could be added to S_U and returns the adding set. R is the set of points that would be removed from S_O; the points in R are greedily selected by traversing all the possibilities.
  - U is the index number for the unsatisfied group, and O is the index for the satisfied group.

I claim that the output set $(S_U \cup E) \cup (S_O \setminus R)$ is a $\frac{c^2}{4}$-approximation solution for the Max-Min Fair Diversification.

Proof. First, note that

$$l^* \geq l^*_{\text{fair}} \tag{1}$$

Since the l^* is the div of the solution in the groupless setting, by applying more constraints, the div of the solution $\mathcal{F}^*$ would be worse. In other words, l^*_{fair} is smaller or equal than l^*.

Then note that

$$\forall i \in S, \quad \left| \{p \in S^* \mid d(p,i) < \frac{cl^*}{2}\} \right| \leq 1$$

This means that **for each** point i in S, there's **at most one point** p in S^* that satisfies: $d(p,i) < \frac{cl^*}{2}$. In other words, there's only one point in S^* such that the distance is $< \frac{cl^*}{2}$ from each point of S. We can prove this by contradiction. Assume that there exist at least two points satisfy (2), then we have $\exists i \in S$, $\exists p, q \in S^*$, s.t. $d(p,i) < \frac{cl^*}{2}$ and $d(q,i) < \frac{cl^*}{2}$.

Then, from triangle inequality and the previous two inequalities, which are

$$c \cdot d(p,q) \leq d(p,i) + d(q,i)$$

$$d(p,i) < \frac{cl^*}{2}, \quad d(q,i) < \frac{cl^*}{2}$$

We have

$$c \cdot d(p,q) \leq d(p,i) + d(q,i) < \frac{cl^*}{2} + \frac{cl^*}{2} = cl^* \implies d(p,q) < l^*$$

Since $l^* = div(S^*)$, which is the Min distance between points in S^*, but now we have points p, q with $d(p,q) < l^*$, which contradicts to the Minimum distance l^*. Then we've proved that for each point in S, there's at most one point in S^* such that $d(p,i) < \frac{cl^*}{2}$.

Therefore, while GMM has picked less than k elements, in the worst case, each point in S has one corresponding point in S^* such that the distance between them $< \frac{cl^*}{2}$ (and different points in S may correspond to the same point in S^* as well), then we can create a mapping between the points in S and the points in S^*. By the Pigeon Hole Principle, since $|S| < |S^*| = k$, there must exist a good point that can be greedily selected and added to S, with distance $\geq \frac{cl^*}{2}$ from all the points already selected. Also, with the fact that the algorithm greedily picks the next point, which is the point farthest away, we can guarantee that the good point that we mentioned would be chosen if there's no other better option. Then, we naturally have

$$div(S) \geq \frac{cl^*}{2} \geq \frac{cl^*_{\text{fair}}}{2} \tag{2}$$

Since S_U is a subset of S, which has fewer points, we observe that

$$div(S_U) \geq div(S) \geq \frac{cl^*_{\text{fair}}}{2}$$

Then we look at set E, which is the set that includes the points that would be added to S_U later. Similar to the previous reasoning, when we are running GMM with parameter $(\mathcal{U}_U, S_U, k_U - |S_U|)$, there is at most

one point in $\mathcal{F}_U^*$ that has distance $< \frac{cl^*_{\text{fair}}}{2}$ from each point in $S_U \cup E$. Formally,

$$\forall i \in S_U \cup E, \quad \left| \{p \in \mathcal{F}_U^* \mid d(p,i) < \frac{cl^*}{2}\} \right| \leq 1$$

Similarly, when GMM has picked less than $k - |S_U|$ elements, there exists at least one element that can be selected with a distance greater or equal to $\frac{cl^*_{\text{fair}}}{2}$ 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 $\frac{cl^*_{\text{fair}}}{2}$ from the existing points. Then, we naturally have

$$div(S_U \cup E) \geq \frac{cl^*_{\text{fair}}}{2} \tag{3}$$

Our output is $(S_U \cup E) \cup (S_O \setminus R)$. From (2) and (3), we already proved that $div(S)$ and $div(S_U \cup E)$ are greater than or equal to $\frac{cl^*_{\text{fair}}}{2}$, the only thing that is left for our proof is $div(E \cup S_O)$. For this case, by the algorithm's logic, we remove the closest point in S_O from E. Note that by application of the triangle inequality and the fact that $div(S_O) \geq \frac{cl^*_{\text{fair}}}{2}$, for each $x \in E$ there can be at most one point $y \in S_O$ such that $d(x,y) < \frac{c^2 l^*_{\text{fair}}}{4}$, and it is obvious that the size of E and R are the same. Hence, after the removal of the closest points, the distances between all pairs are as required, which completes the proof. □

5 Proof for 5-approximation results for extended metric space

This section is about the extended metric space version of the algorithm proposed in Section 3.1.2, Figure 4. This is only a draft section; some parts of the proof still need more time to refine. I will highlight the parts that need refinement in blue for indication purposes. We recommend the reader to have a look at the previous section and algorithm pseudo-code for a better understanding of the proof.

Theorem 6. *Fair-GMM is a* 5*-approximation algorithm for the fair diversification problem with extended metric factor* $c \in (0, 2]$*, and the running time is* $O(kn)$

The Theorem 6 proof here is only for $c \in [1, 2]$ rather than $c \in (0, 2]$. In terms of the case $c \in (0, 1]$, it would be the same as the original proof from Moumoulidou et al. [2020], due to the stricter triangle inequality. However, we might improve the case of $c \in (0, 1]$ with a better approximation ratio in the future.

Proof. In the context of the algorithm described by the pseudo-code (Figure 4), consider the union of sets $Y_1 \cup \cdots \cup Y_m$. Our objective is to demonstrate that, upon conducting an exhaustive search on this union, we can identify a solution set $S = S_1 \cup \cdots \cup S_m$. This solution set will satisfy the condition $div(S) \geq \frac{l^*}{5}$. Here, l^* represents an optimal solution value, and c denotes the factor associated with the extended metric space. Consequently, this leads to an approximation ratio of 5.

Define Z_i as the largest subset within Y_i such that Z_i fulfills the condition $div(Z_i) \geq \frac{2cl^*}{5}$.

For each $x \in U_i$, let $f(x)$ be the closest point in Z_i.

We call a group critical if $|Z_i| < k$, which holds the following property:

Remark 2. If a group i is critical (i.e.$|Z_i| < k$), then $\forall x \in U_i, d(x, f(x)) < \frac{2cl^*}{5}$

To prove Remark 2, first note that $|Z_i| < k$ implies that we exclude some points from Y_i, because we exclude the point from Z_i only if the point is too close to others (i.e., distance $< \frac{2cl^*}{5}$ from other points in the group). In the other word, if group i is critical, then $div(Y_i) < \frac{2cl^*}{5}$. Then, we can prove by cases.

1. $x \in U_i \wedge x \in Z_i$: In this case, by definition of $f(x)$ we have $f(x) = x$. $d(x, x) = 0 < \frac{2cl^*}{5}$
2. $x \in U_i \wedge x \in (Y_i \setminus Z_i)$: Since x is included in Y_i but not Z_i, by our definition of Z_i, this case is trivially satisfied.
3. $x \in U_i \setminus (Y_i \cup Z_i)$: this is the part left for future proof

Therefore, we have proved: If a group i is critical (i.e.$|Z_i| < k$), then $\forall x \in U_i, d(x, f(x)) < \frac{2cl^*}{5}$.

Let O_i be the optimal set of group i. Consider the subset $S_1, S_2, \ldots, S_m$ of $Z_1, Z_2, \ldots, Z_m$, which is defined as follows:

- For Critical Groups: Let $S_i = f(O_i)$ and let $D = \bigcup_{i:critical} S_i$. Note that:
  - $|D| = \sum_{i:critical} k_i$: trivially satisfied.
  - $div(D) > \frac{l^*}{5}$:

    To prove this, we consider two cases:

    * Same group: $x, y \in S_i$, then $d(x, y) \geq \frac{2cl^*}{5} \geq \frac{l^*}{5}$ by definition.
    * Cross group: Consider x and y in O_i and O_j, with the corresponding $f(x)$ and $f(y)$ in S_i and S_j. Since x, y are in the optimal group, then $d(x, y) \geq l^*$. By Remark 2, we have $d(x, f(x)) < \frac{2cl^*}{5}$ and $d(y, f(y)) < \frac{2cl^*}{5}$.

      I would like to introduce a remark here to assist the rest of the proof.

      Remark 3. Let $\triangle ABC$ and $\triangle ACD$ be the triangles in the extended metric space with factor c, then $BD \geq AC - (AB + CD)/c$.

      The proof of the remark is stated here.

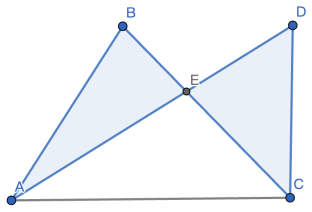

Figure 11: Triangle

Proof. Let $\triangle ABC$ and $\triangle ACD$ be the triangles in the extended metric space with factor c.

Then, by the triangle inequality in the extended metric space, we have

$$c(BD + AB) \geq AD$$
$$c(AC + AB) \geq BC$$

By taking the difference between these two inequalities, we have

$$\begin{aligned} c(BD - AC) &\geq (AD - BC) \\ c \cdot BD &\geq AD - BC + c \cdot AC \\ &\geq c \cdot AC + (AE + DE) - (BE + CE) \\ &\geq c \cdot AC + (AE - BE) + (DE - CE) \qquad (*) \end{aligned}$$

Then, I want to show that $AE - BE \geq -AB$ and, by symmetry, $DE - CE \leq -CD$.

$$\begin{aligned} c(AE + AB) &\geq BE \\ (AE + AB) &\geq BE/c \\ AB &\geq BE/c - AE \geq BE - AE \qquad \text{require } c \in [1, 2) \\ AE - BD &\geq -AB \end{aligned}$$

Similarly, we have $DE - CE \leq -CD$. Then plug the results into $(*)$

$$\begin{aligned} c \cdot BD &\geq c \cdot AC + (AE - BE) + (DE - CE) \\ &\geq c \cdot AC - AB - CD \\ BD &\geq AC - (AB + CD)/c \end{aligned}$$

Here, we've proved what we want. □

Then since we are interested in cross group min distance, which means that we are interested in BD in the previous proof. By the optimal set definition, Remark 2 and Remark 3 we have the following inequalities.

$$div(O_i) \geq l^*$$
$$d(x, f(x)) < \frac{2cl^*}{5}, d(y, f(y)) < \frac{2cl^*}{5}$$
$$d(f(x), f(y)) \geq div(O_i) - d(x, f(x))/c - d(y, f(y))/c$$

From the above inequalities, we can conclude

$$d(f(x), f(y)) > l^* - \frac{4l^*}{5} > \frac{l^*}{5}$$

for $c \in [1, 2]$ where $f(x), f(y) \in D$ by our setup. Then $div(D) > \frac{l^*}{5}$ property for D is proved fully.

- Then, let us focus on the j that is not critical. For each group j that is not critical, remove the point

from Z_j that is distance $< \frac{l^*}{5}$ from the point in D. We claim that there are at most $|D| = |\sum_{i:critical} k_i|$ points removed from Z_j. Because for each point p_D in D, there is at most 1 point p_j from Z_j s.t. $d(p_D, p_j) < \frac{l^*}{5}$.

We would like to prove the statement above. Formally, the statement is

$$\forall p_D \in D, \left| \{p_j \in Z_j | d(p_j, p_D) < \frac{l^*}{5}\} \right| \leq 1$$

Proof. Prove the statement by contradiction.

Let $p_D \in D$, such that there exist two points p_1, p_2 in Z_j, $d(p_1, p_D) < \frac{l^*}{5}$ and $d(p_2, p_D) < \frac{l^*}{5}$

By triangle inequality, we have

$$c(d(p_1, p_D) + d(p_2, p_D)) \geq d(p_1, p_2)$$

Because $div(Z_j) \geq \frac{2cl^*}{5}$, then $d(p_1, p_2) \geq \frac{2cl^*}{5}$. While by $d(p_1, p_D) < \frac{l^*}{5}$ and $d(p_2, p_D) < \frac{l^*}{5}$, we have $c(d(p_1, p_D) + d(p_2, p_D)) < \frac{2cl^*}{5}$.

Then, we need to make both of the following inequalities valid, which is not possible and leads to the contradiction:

$$c(d(p_1, p_D) + d(p_2, p_D)) \geq d(p_1, p_2) \geq \frac{2cl^*}{5}$$

$$c(d(p_1, p_D) + d(p_2, p_D)) < \frac{2cl^*}{5}$$

□

- For each non-critical j, we generate the S_j by following steps in a arbitrary order:
  - Pick k_j points randomly from Z_j to form S_j
  - In $Z = \bigcup_{i:non-critical} Z_i$, remove the points which has distance $< \frac{l^*}{5}$ from a point in S_j. This would remove at most k_j points from each Z_i. When we process j, there are at least $(k - \sum_{i \in [m], i \neq j} k_i)$ points, which is k_j points in Z_j for choosing.

□

Here is a side note that I would like to tell the reader about this section and proof:

I didn't find the place that requires $2l^*/5$ in the proof. It seems like we can even replace this divergence with $2l^*/5$. If we can do it, this algorithm will be proved as a 3-approximation algorithm.

The guess for why we need $2l^*/5$ specifically is that we might need to use this in Remark 2. However, as I indicated in the proof of Remark 2, I cannot prove the remark statement's correctness for Case 3. That would be the key part I will work on in the near future.

The other doubt of this algorithm is that we are asked to choose k points for each group and remove $k - k_i$ points to satisfy the fairness constraints, but what if there's a group with less than k points? It would be

a challenge for our fairness constraint because our proof only shows that we **at most** remove $k - k_i$ points, which means that we may left with $< k_i$ points in the solution for the group with $< k$ points.

In terms of future work, except the refinement of the proof of Remark 2 and thinking about the group with less than k points, I would also try to work on a BETTER result for $c \in (0, 1]$. We can directly have the 5-approximation result by slightly modifying the original proof(I don't have time to write it down, but because it is a stricter triangle inequality, all results that applied to the normal triangle inequality $a + b \geq c$ should be inherited, for this stricter case.)

6 Improvements of 6-Approximation Algorithm

In this section, we propose two enhancements to the 6-approximation algorithm presented in Section 3.2.2. Firstly, we introduce a constant optimization in the selection of the parameter ϵ. This refinement aims to enhance the algorithm's accuracy while maintaining its efficiency. Secondly, we address the issue of $1 - \epsilon$ fairness by proposing an adjustment that achieves perfect fairness by increasing the input size.

6.1 Constant Improvement for ϵ

The context for this section is given in Section 3.2.2

Remark 4. Let $\epsilon \geq \sqrt{\frac{2\log(2m)}{k_i}}$ for all $i \in [m]$, $\delta \in \mathbb{R}$, we have

$$\Pr[\sum_{p \in \mathcal{X}_i} Y_p \leq (1-\epsilon)k_i] \leq \exp\left(-\epsilon^2 k_i/2\right) \leq \frac{1}{2m}$$

Proof. By the Chernoff Bound Definition that is given (Definition 10) and adopting our variable definition, we have

$$P[\sum_{p \in \mathcal{X}_i} Y_p \leq (1-\epsilon)k_i] \leq \left(\frac{e^{-\epsilon}}{(1-\epsilon)^{1-\epsilon}}\right)^{k_i}$$

Let $c \in \mathbb{R}$, we would like to find the smallest value of c for the following inequality:

$$P[\sum_{p \in \mathcal{X}_i} Y_p \leq (1-\epsilon)k_i] \leq \left(\frac{e^{-\epsilon}}{(1-\epsilon)^{1-\epsilon}}\right)^{k_i} \leq \exp(-\frac{\epsilon^2}{c}k_i)$$

We would like to show that the smallest value c is equal to 2 here.

$$\left(\frac{e^{-\epsilon}}{(1-\epsilon)^{1-\epsilon}}\right)^{k_i} \leq \exp(-\frac{\epsilon^2}{c}k_i)$$
$$(-\epsilon-(1-\epsilon)\log(1-\epsilon))k_i \leq -\frac{\epsilon^2}{c}k_i$$
$$-\epsilon-(1-\epsilon)\log(1-\epsilon) \leq -\frac{\epsilon^2}{c}$$
$$\frac{\epsilon^2}{\epsilon+(1-\epsilon)\log(1-\epsilon)} \leq c$$

In order to find the smallest c that greater or equal to $\frac{\epsilon^2}{\epsilon+(1-\epsilon)\log(1-\epsilon)}$ for every $\epsilon \in (0,1)$, first we can observe that $\frac{\epsilon^2}{\epsilon+(1-\epsilon)\log(1-\epsilon)}$ is a decreasing monotonic function. Therefore, for domain $\epsilon \in (0,1)$, we get the local maximum when $\epsilon \to 0$, which would be our best choice of c.

Let c^* be the optimal c. We have

$$\begin{aligned}
c^* &= \lim_{\epsilon\to 0}\frac{\epsilon^2}{\epsilon+(1-\epsilon)\log(1-\epsilon)} \\
&= \lim_{\epsilon\to 0}\frac{\frac{d}{d\epsilon}\epsilon^2}{\frac{d}{d\epsilon}(\epsilon+(1-\epsilon)\log(1-\epsilon))} \qquad \text{(L'Hôpital's rule)} \\
&= \lim_{\epsilon\to 0}-\frac{2\epsilon}{\log(1-\epsilon)} \\
&= -2\lim_{\epsilon\to 0}\frac{\epsilon}{\log(1-\epsilon)} \\
&= -2\lim_{\epsilon\to 0}\frac{\frac{d}{d\epsilon}\epsilon}{\frac{d}{d\epsilon}\log(1-\epsilon)} \qquad \text{(L'Hôpital's rule)} \\
&= -2\lim_{\epsilon\to 0}\frac{1}{-\frac{1}{1-\epsilon}} = 2\lim_{\epsilon\to 0}(1-\epsilon) \\
&= 2
\end{aligned}$$

□

So after our proof, Theorem 2 could be improved to:

Theorem 7. *Let $\epsilon \geq \sqrt{\frac{2\log(2m)}{k_i}}$ for all $i \in [m]$, $\delta \in \mathbb{R}$. There is a poly(n, k, δ^{-1}) time algorithm that returns a subset of points with diversity $l^*/6$ and includes $(1-\epsilon)k_i$ points in each group $i \in [m]$ with probability at least $(1-\delta)$ if we run the algorithm $\log \delta^{-1}$ times.*

6.2 Improvement from $1-\epsilon$ Fairness to Perfect Fairness

In order to improve the fairness from $1-\epsilon$ to perfect, prior to applying our main algorithm to the real input, it is essential to slightly adjust the input. Originally, for each group, we select points $k_1, \ldots, k_m$. We propose

modifying these to $k'_1 = \frac{k_1}{1-\epsilon}, \ldots, k'_m = \frac{k_m}{1-\epsilon}$.

This adjustment is both valid and effective. The validity stems from the only constraint we have is ϵ, which is $\epsilon \geq \sqrt{(3\log(2m)/k_i)}$, and has been further refined to $\epsilon \geq \sqrt{(2\log(2m)/k_i)}$ as per our previous subsection. With a fixed ϵ determined by the initial values of $k_1, \ldots, k_m$, increasing k_i to k'_i maintains the constraint's validity.

Then, we would show that the modification is effective in improving fairness. From the algorithm, inputting $k'_1, \ldots, k'_m$ results in a $1-\delta$ probability of giving a set where $|S \cap \mathcal{X}_i| = (1-\epsilon)k'_i$ for all groups. Given that $k'_i = \frac{k_i}{1-\epsilon}$, this essentially equates to obtaining a set with $|S \cap \mathcal{X}_i| = k_i$, which ensures the perfect fairness across all groups. This modification effectively enhances fairness from a $1-\epsilon$ level to perfect. Additionally, if the solution set contains excess points, they can be efficiently pruned using a greedy approach, potentially further improving the $div(S)$, i.e., the diversity of the solution set.

7 Investigation of FairSwap Algorithm in $m \geq 3$ cases

The algorithm outlined in Section 4 primarily addresses settings involving two distinct groups. A straightforward approach to extend its application to scenarios with more than two groups might involve the divide-and-conquer method. However, this strategy is impractical for the algorithm in question due to the absence of optimal substructure within the problem. This means that achieving optimal solutions for individual subgroups does not guarantee the feasibility of merging these solutions into a single, optimal solution.

8 Conclusion and Open Questions

8.1 Conclusion

In this study, we have improved the results upon earlier findings related to the Fair Max-Min Diversification Problem within a metric space; we have also extended some earlier results to an extended metric space. We achieved a $\frac{4}{c^2}$-approximation for this problem in extended metric space when $m = 2$ by applying the FairSwap algorithm to this new setting. Our exploration included an attempt to generalize the FairSwap Algorithm into the $m \geq 3$ case, although this endeavor was unsuccessful. Additionally, we extended the 5-approximation results for cases with small values of k and m in the extended metric space. Beyond these findings, our work has further refined the 6-approximation algorithm in two ways. Firstly, we have improved the constant of ϵ. Secondly, we have enhanced the fairness measure, progressing from $1-\epsilon$ to perfect fairness. This advancement was achieved by strategically manipulating the input in a valid manner.

8.2 Open Questions

Future Directions for the Topic

Current research, as discussed in Moumoulidou et al. [2020], suggests the negative result of the approximation ratio; we cannot have an algorithm with an approximation ratio better than 2 for Fair Max-Min Diversification problem. The state-of-the-art results with perfect fairness include a 4-approximation algorithm with perfect fairness in the $m = 2$ scenario and an $m + 1$-approximation algorithm for more general cases. However, there exists an opportunity to narrow this gap, either by refining the hardness proof or by developing new algorithms.

Explorations in extended metric spaces present another possible direction. Several existing algorithms could potentially be adapted to this expanded context.

Another consideration, as highlighted in Addanki et al. [2022], is the extension of fairness constraints to arbitrary matroid constraints. Though there are known results for other diversity maximization problems under matroid constraints (as cited in references Abbassi et al. [2013], Bhaskara et al. [2016], and Borodin et al. [2012]), further generalizations remain an open area for the Max-Min Diversification problem.

Near-Term Work from a Personal Perspective

In addition to the aforementioned directions, there's some work left for this paper which needs some time to finalize.

- Finalize the 5-approximation proof for extended metric spaces, and attempt to achieve better outcomes for $c \in (0, 1]$.
- Investigate the $m + 1$-approximation algorithm with perfect fairness in extended metric spaces, especially for overlapping groups. The anticipated approximation ratio is $\begin{pmatrix} m \\ \lfloor m/2 \rfloor \end{pmatrix} + 1$.
- Moumoulidou et al. [2020] suggest extending the FairSwap (2-approximation algorithm) to general metrics. Although addressed in Section 7, further exploration is needed to assess if it is feasible to bypass the merging process or achieve a valid approximation that is potentially worse than 2 for $m \geq 3$.

9 Appendix

(Note: In the previous section, we used c as the extended metric space factor, while in this appendix, α is considered as the factor for the metric space. They are the same thing under different notations.)

We cannot have $\alpha > 2$ for the factor of extended metric space. In addition, if $\alpha = 2$, it enforces that all non-zero distances in such metric to become identical. Notice that the problem of interests would be trivial if the distances are identical. I will give proof for α's range first and then show that the distances are identical when $\alpha = 2$.

Here is a simple proof for the statement to show why $\alpha = 2$ is the maximum value for α.

Proof. Let $\alpha \in \mathbb{R}^+$

We have

$$\begin{aligned} a+b \geq \alpha c,\ \ b+c \geq \alpha a,\ \ a+c \geq \alpha b \\ \implies a+b \geq \alpha(\alpha a - b) \\ \implies a+b \geq \alpha^2 a - \alpha b \\ \implies (1+\alpha)b \geq (\alpha^2-1)a \end{aligned}$$

We also have $(1+\alpha)a \geq (\alpha^2-1)b$

Then we have

$$\left(\frac{1+\alpha}{\alpha^2-1}a\right) \geq b$$
$$\left(\frac{1+\alpha}{\alpha^2-1}b\right) \geq a$$

We need to have $\frac{1+\alpha}{\alpha^2-1} \geq 1$ to make this possible, regardless of the value of a, b. Then

$$1+\alpha \geq \alpha^2 - 1 \implies \alpha \leq 2$$

Then we reach our conclusion where 2 is the maximum value for the α. □

Here is a justification where we would have identical distance when $\alpha = 2$:

Proof. Assume that $\alpha = 2$, the distances between three points are a, b, c.

From this setup, we have $a+b \geq 2c,\ \ a+c \geq 2b,\ \ b+c \geq 2a$.

Then $a+b \geq 2c \geq 2(2b-a) \implies a+b \geq (4b-2a) \implies 3a \geq 3b$

Similarly, we get $3b \geq 3a$, then we have $a = b$. By symmetry, $a = b = c$. The distances in this metric are identical. □

10 Acknowledgement

I would like to thank Professor Allan Borodin for their guidance and support throughout this project. Your advice and expertise have been invaluable.

References

Z. Abbassi, V. S. Mirrokni, and M. Thakur. Diversity maximization under matroid constraints. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, page 32–40, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450321747. doi: 10.1145/2487575.2487636. URL https://doi.org/10.1145/2487575.2487636.

R. Addanki, A. McGregor, A. Meliou, and Z. Moumoulidou. Improved approximation and scalability for fair max-min diversification, 2022.

A. Bhaskara, M. Ghadiri, V. Mirrokni, and O. Svensson. Linear relaxations for finding diverse elements in metric spaces. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4105–4113, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

A. Borodin, H. C. Lee, and Y. Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS '12, page 155–166, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450312486. doi: 10.1145/2213556.2213580. URL https://doi.org/10.1145/2213556.2213580.

Z. Moumoulidou, A. McGregor, and A. Meliou. Diverse data selection under fairness constraints, 2020.

S. Ravi, D. Rosenkrantz, and G. Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42:299–310, 04 1994. doi: 10.1287/opre.42.2.299.

D. Wang and Y.-S. Kuo. A study on two geometric location problems. *Information Processing Letters*, 28 (6):281–286, 1988. ISSN 0020-0190. doi: https://doi.org/10.1016/0020-0190(88)90174-3. URL https://www.sciencedirect.com/science/article/pii/0020019088901743.