Towards Fair Representation: Clustering and Consensus

Diptarka Chakraborty DIPTARKA@COMP.NUS.EDU.SG
National University of Singapore

Kushagra Chatterjee KUSHAGRA.CHATTERJEE@U.NUS.EDU
National University of Singapore

Debarati Das DXD5606@PSU.EDU
Pennsylvania State University

Tien Long Nguyen TFN5179@PSU.EDU
Pennsylvania State University

Romina Nobahari NOBAHARIROMINA@GMAIL.COM
Sharif University of Technology

Editors: Nika Haghtalab and Ankur Moitra

Abstract

Consensus clustering, a fundamental task in machine learning and data analysis, aims to aggregate multiple input clusterings of a dataset, potentially based on different non-sensitive attributes, into a single clustering that best represents the collective structure of the data. In this work, we study this fundamental problem through the lens of fair clustering, as introduced by Chierichetti et al. [NeurIPS'17], which incorporates the disparate impact doctrine to ensure proportional representation of each protected group in the dataset within every cluster. Our objective is to find a consensus clustering that is not only representative but also fair with respect to specific protected attributes. To the best of our knowledge, we are the first to address this problem and provide a constant-factor approximation.

As part of our investigation, we examine how to minimally modify an existing clustering to enforce fairness – an essential postprocessing step in many clustering applications that require fair representation. We develop an optimal algorithm for datasets with equal group representation and near-linear time constant factor approximation algorithms for more general scenarios with different proportions of two group sizes. We complement our approximation result by showing that the problem is NP-hard for two unequal-sized groups. Given the fundamental nature of this problem, we believe our results on Closest Fair Clustering could have broader implications for other clustering problems, particularly those for which no prior approximation guarantees exist for their fair variants.

Keywords: Fairness, Consensus Clustering, Closest Fair Clustering, Approximation Algorithms.

1. Introduction

Machine learning plays a crucial role in modern decision-making processes, including recommendation systems, economic opportunities such as loan approvals, and recidivism prediction, e.g. Kleinberg et al. (2016, 2018). However, these machine learning-driven processes risk being biased against marginalized communities based on sensitive attributes, such as gender or race Kay et al. (2015); Bolukbasi et al. (2016), due to biases inherent in the training data. This highlights the need to study and design fair algorithms to address disparities resulting from historical marginalization. In recent years, there has been significant literature on algorithmic fairness, focusing on achieving *demographic parity* Dwork et al. (2012) or *equal opportunity* Hardt et al. (2016).

Clustering is a foundational unsupervised learning task that involves partitioning a set of data points in a metric space into clusters, with the goal of minimizing certain objective functions specific to the application. Each data point can be viewed as an individual with certain protected attributes, which can be represented by assigning a color to each point. Chierichetti et al. (2017) pioneered the concept of *fair clustering*, where each point in the dataset is colored either red or blue. The goal was to partition the data while maintaining balance in each cluster – the ratio of blue to red points in each cluster reflects the overall ratio in the entire set. This balance is important for addressing disparate impact and ensuring fair representation. Since it was introduced, different variants of fair clustering have been studied, including k-center/median/means clustering Chierichetti et al. (2017); Huang et al. (2019), scalable clustering Backurs et al. (2019), proportional clustering Chen et al. (2019), overlapping multiple groups Bera et al. (2019), group-oblivious models Esmaeili et al. (2020), correlation clustering Ahmadian et al. (2020); Ahmadi et al. (2020); Ahmadian and Negahbani (2023) and 1-clustering over rankings Wei et al. (2022); Chakraborty et al. (2022), among others.

In this paper, we consider *consensus clustering*, a fundamental problem in machine learning and data analysis, and initiate its study under fairness constraints. Given a set of clusterings over a dataset, potentially based on different non-sensitive attributes, the goal of consensus clustering is to aggregate them into a single clustering that best represents the collective structure of the data while minimizing a specified objective function. The choice of objective functions varies depending on the context, with two of the most common being the *median objective* that minimizes the sum of distances, and the *center objective* that minimizes the maximum distance. Here, the distance between two clusterings is measured by counting the pairs of points that are grouped together in one clustering but not in the other (see Section 2 for a formal definition). The consensus clustering problem has a myriad of applications in different domains, such as gene integration in bioinformatics Filkov and Skiena (2004b,a), data mining Topchy et al. (2003), community detection Lancichinetti and Fortunato (2012). The problem (both respect to median and center objective) is not only known to be $\mathtt{NP}$-hard Křivánek and Morávek (1986); Swamy (2004), but also known to be $\mathtt{APX}$-hard, i.e., unlikely to possess any $(1+\epsilon)$-approximation (for any $\epsilon > 0$) algorithm even when there are only three input clusterings Bonizzoni et al. (2008). While several heuristics have been considered to generate reasonable solutions (e.g., Goder and Filkov (2008); Monti et al. (2003); Wu et al. (2014)), so far, we only know of an $11/7$-approximation algorithm for the median objective Ailon et al. (2008), and a better than 2-approximation for the center objective Das and Kumar (2025).

Unfortunately, the consensus algorithms mentioned above do not account for fairness in clustering. For example, take the task of detecting community structures within a social network. There may be various valid partitions available based on users' non-sensitive attributes, such as age group or food preferences. However, when creating an overall community partition, it is important for each community to be fair, meaning each group should be fairly represented according to specific protected attributes like race or gender. Specifically, we refer to the concept of fair clustering as introduced by Chierichetti et al. (2017), where a clustering of red-blue points is deemed fair if each cluster individually reflects the population ratio of red and blue points. It raises a key computational question: given a set of clusterings, how can we derive a fair clustering that effectively aggregates the input clusterings? To our knowledge, the current work is the first to investigate this computational fair aggregation challenge.

En route, we consider another innate, perhaps even more fundamental, computational question concerning fairness in clustering – to transform an arbitrary clustering into its nearest fair clustering. This question is intriguing on its own merit. We might have an effective clustering algorithm for specific tasks, but due to potential biases in the training data, the result might not be fair, even if it is a good clustering of the input. In such cases, we seek to achieve a fair clustering by making minimal changes to the output clusters. This type of postprocessing is evidently essential in most clustering applications because it is imperative to ensure fairness among the clusters produced. In many scenarios, clustering algorithms can inadvertently result in biased groupings that do not adequately represent all protected classes or groups within the dataset. Such skewed representations can lead to unfair treatment or outcomes, particularly when the clusters are used for decision-making or analytical purposes. Therefore, to prevent these biases and promote equity, it is necessary to apply postprocessing techniques that adjust the clusters to reflect a fair distribution of the protected attributes. This ensures that each group is proportionately represented and that the clustering results do not inadvertently favor one group over another, thereby upholding principles of fairness and non-discrimination in the analysis. Similar fairness-related questions have been examined in other contexts, such as ranking candidates Celis et al. (2018); Chakraborty et al. (2022); Kliachkin et al. (2024). Surprisingly, this computational task has not garnered significant attention within the broader clustering literature. In this study, we also address this fundamental question of fair clustering. As we will demonstrate later, this question is closely linked to the challenge of creating a fair consensus.

1.1. Our Contribution

Closest Fair Clustering We begin by introducing the study of *Closest Fair Clustering* – the problem of modifying an arbitrary input clustering to achieve fairness with the minimal number of changes. The input dataset consists of points having a color, either red or blue. Starting with an arbitrary clustering of this colored data set, the goal is to identify the nearest/closest fair clustering. First, we study the case where the two colored subsets are of equal size and propose an optimal algorithm for finding the closest fair clustering in this setting.

Theorem 1 *There is an algorithm that, given a clustering over a set of n red-blue colored points with an equal number of red and blue points, outputs a closest* Fair clustering *in $O(n \log n)$ time.*

Next, we explore a more general scenario where the data set is not perfectly balanced; specifically, the ratio of blue to red points is an arbitrary constant. Due to the symmetry between blue and red points, we assume, without loss of generality, that this ratio is at least one throughout this paper. It is important to note that fair versions of various clustering problems turn out to be NP-hard for such arbitrary ratios, particularly when the ratio exceeds a certain (small) constant greater than one, as demonstrated in Chierichetti et al. (2017). However, that does not imply that the closest fair clustering problem is also NP-hard. We show that the closest fair clustering problem is NP-hard even for any integral ratio strictly greater than one (see the full version[1]). We establish the NP-hardness result by providing a reduction from the 3-*Partition* problem.

Next, we design a near-linear time algorithm that provides a constant-factor approximation for the closest fair clustering problem for arbitrary integral ratios. We refer to α-approximation as α-*close*, for $\alpha \geq 1$ (see Section 2 for a formal definition).

Theorem 2 *Consider an integer $p > 1$. There is an algorithm that, given a clustering over a set of n red-blue colored points where the ratio between the total number of blue and red points is p, outputs a 17-close* Fair clustering *in time $O(n \log n)$.*

[1] The full version of this paper is available at https://arxiv.org/abs/2506.08673.

To demonstrate the aforementioned result, we employ a two-stage approach. In the first step, we ensure that each input cluster becomes *balanced* – the size of the blue subpart in each cluster is a multiple of p. We develop a 3.5-approximation algorithm for this purpose. Subsequently, we introduce an approximation algorithm that transforms any balanced clustering into a fair clustering that is approximately close, more specifically, 3-close. By integrating these two steps, we achieve a 17-approximation guarantee, as stated in the theorem above.

The closest fair clustering problem becomes even more complex when the population ratio between blue and red points is fractional (represented as p/q for two coprime numbers p and q). It is worth remarking that the closest fair clustering problem is NP-hard for any arbitrary ratio strictly greater than one. To design an approximation algorithm, we apply the previously mentioned two-stage approach again. Since the second step is effective regardless of whether the ratio is integral or fractional, we can utilize it unchanged. Our attention must now shift to the initial balancing step, where we need to ensure that the blue subpart of each cluster is a multiple of p and the red subpart is a multiple of q.

A straightforward approach to achieve balancing is to apply the balancing step for the integral ratio case twice sequentially: first, to make blue subparts a multiple of p and then to make red subparts a multiple of q. We can then argue that an α-approximation balancing algorithm for the integral case results in a $(\alpha^2 + 2\alpha)$-approximate balancing algorithm for the fractional case. This yields a 19.25-approximation just for the balancing step, given that our integral balancing achieves a 3.5-approximation. Therefore, combining both the balancing and the process of achieving a fair clustering from a balance clustering results in an 80-approximation factor for the closest fair clustering.

However, we propose a more sophisticated balancing algorithm and, through an intricate analysis, manage to avoid the squared increase in the approximation factor, achieving an approximation factor of only 7.5. Finally, by combining both stages, we develop a 33-approximation algorithm, significantly improving upon the naive bound of an 80-approximation.

Theorem 3 *Consider two integers $p, q > 1$. There is an algorithm that, given a clustering over a set of n red-blue colored points where the ratio between the total number of blue and red points is p/q, outputs a 33-close Fair clustering in time $O(n \log n)$.*

Closest Fair Clustering using Fair Correlation Clustering. It is worth emphasizing the connection between the closest fair clustering and the fair correlation clustering. In correlation clustering, we are given a labeled complete graph where each edge is labeled either $+$ or $-$. The cost of a clustering (of the nodes) is defined as the summation of the number of intercluster $+$ edges and the intracluster $-$ edges. The *fair correlation clustering* problem asks to find a fair clustering with minimum cost. It is easy to observe that given a clustering (on n points), we can find its closest fair clustering by solving the fair correlation clustering problem on the following instance: Create a complete graph on n nodes, where an (undirected) edge (u, v) is labeled $+$ if both u, v belong to the same cluster in the input clustering; otherwise it is labeled $-$.

Due to the above reduction, by deploying the current state-of-the-art approximation algorithms for the fair correlation clustering, we immediately get approximation algorithms for the closest fair clustering problem; however, this leads to a much worse approximation factor. The correlation clustering problem (even on complete graphs) is already NP-hard (in fact, APX-hard), and its fair variant remains NP-hard even when the blue-to-red ratio is one Ahmadi et al. (2020). In contrast, we provide an exact algorithm for the closest fair clustering problem when the ratio is one (Theorem 1).

Thus, we, in fact, get a separation between these two problems – the closest fair clustering problem and the fair correlation clustering problem – for the ratio of one. If we had used the state-of-the-art fair correlation clustering algorithm Ahmadian et al. (2020); Ahmadi et al. (2020), we would have only obtained an $O(1)$ approximation factor for the closest fair clustering problem in this case.

For arbitrary ratios, even when the ratio is an integer p, the state-of-the-art fair correlation clustering algorithms Ahmadian et al. (2020); Ahmadi et al. (2020) give us an approximation factor of $O(p^2)$. In contrast, we get an approximation factor of 17 (Theorem 2) for the closest fair clustering problem. Note that the approximation factor obtained by our algorithm is entirely independent of the blue-to-red ratio (i.e., independent of p). Further, due to the hidden constant of $O(\cdot)$ notation in the approximation factor of the algorithms of the fair correlation clustering Ahmadian et al. (2020); Ahmadi et al. (2020), the implied approximation bound is a much larger constant even for $p = 2$ compared to our algorithm.

Fair Consensus Clustering Next, we focus on the challenge of combining multiple input clusterings into a single clustering while ensuring fairness in the resulting output. We address this fair consensus clustering problem under a generalized aggregation objective function, known as the *generalized mean objective* or simply the *ℓ-mean objective*[2]. This generalized mean objective encompasses a variety of classical optimization objectives, ranging from the *median* when $\ell = 1$ to the *center* when $\ell = \infty$, and it has attracted considerable interest in the broader clustering literature Chlamtác et al. (2022). The consensus clustering problem, even without fairness constraints, is APX-hard Bonizzoni et al. (2008). When incorporating fairness constraints, the problem becomes only harder, making it inevitable to incur a constant (multiplicative) approximation factor.

We introduce a generic approximation algorithm for fair consensus clustering by leveraging the solution to the closest fair clustering as a foundational element. Notably, our approach to fair consensus clustering is effective regardless of the parameter ℓ in the generalized mean objective. As a result, it is applicable to both the median objective (when $\ell = 1$) and the center objective (when $\ell = \infty$), as well as other more generalized objectives within this spectrum.

Using the result of Theorem 1 together with our generic fair consensus approximation algorithm, we achieve a 3-approximation to the fair consensus clustering for the case when the whole population is perfectly balanced.

Theorem 4 *Consider any $\ell \geq 1$. There is an algorithm that, given a set of m clusterings each over a set of n red-blue colored points with an equal number of red and blue points, outputs a 3-approximate ℓ-mean fair consensus clustering in time $O(m^2n^2)$.*

Next, by applying the result from Theorem 2 in conjunction with our general fair consensus approximation algorithm, we attain a 19-approximation for fair consensus clustering in scenarios where the entire population has an integral ratio of blue to red points.

Theorem 5 *Consider any $\ell \geq 1$ and an integer $p > 1$. There is an algorithm that, given a set of m clusterings each over a set of n red-blue colored points where the ratio between the total number of blue and red points is p, outputs a 19-approximate ℓ-mean fair consensus clustering in time $O(m^2n^2)$.*

[2] In the existing literature, this exponent parameter is usually referred to as q, but we use ℓ here to avoid potential confusion with the parameters representing the blue-to-red group size ratio.

Finally, by utilizing the result from Theorem 3 along with our general fair consensus approximation algorithm, we obtain a 35-approximation for fair consensus clustering when the entire population exhibits an arbitrary fractional ratio of blue to red points.

Theorem 6 *Consider any $\ell \geq 1$ and two integers $p, q > 1$. There is an algorithm that, given a set of m clusterings each over a set of n red-blue colored points where the ratio between the total number of blue and red points is p/q, outputs a 35-approximate ℓ-mean fair consensus clustering in time $O(m^2 n^2)$.*

We summarize our main results below.

Clustering Problem	Perfectly Balanced $(1:1)$	Integral Ratio $(p:1)$	Fractional Ratio $(p:q)$
Closet Fair	Optimal (Theorem 1)	17-approx. (Theorem 2)	33-approx. (Theorem 3)
Fair Consensus	3-approx. (Theorem 4)	19-approx. (Theorem 5)	35-approx. (Theorem 6)

Table 1: Table summarizing main results

2. Preliminaries

Notations. In this paper, we consider clusterings over a set V of red-blue colored points. We use n to denote the size of V. Throughout this paper, we consider the points in V are colored either red or blue. For any subset of points $S \subseteq V$, we use $\mathit{red}(S)$ (resp. $\mathit{blue}(S)$) to denote the subset of all red (resp. blue) points in S. Further, without loss of generality, we assume that the number of blue points in the whole set V is at least that of red points, i.e., $|\mathit{blue}(V)| \geq |\mathit{red}(V)|$. For positive integers a, b, p, we use $a \overset{p}{\equiv} b$ to denote that a is congruent to b under modulo p.

Consensus clustering. Between two clusterings $\mathcal{M}, \mathcal{C}$, we define their distance, denoted by $\mathit{dist}(\mathcal{M}, \mathcal{C})$, to be the number of unordered pairs $i, j \in V$ that are clustered together by one but separated by another. The consensus clustering problem asks to aggregate a set of clustering over the points V. In this paper, we consider consensus clustering with respect to a generalized objective function.

Definition 7 (Generalized Mean Consensus Clustering) *Consider an exponent parameter $\ell \in \mathbb{R}$. Given a set of m clusterings $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m$ over a point set V, the ℓ-*mean Consensus Clustering *problem asks to find a clustering $\mathcal{C}$ (not necessarily from the set of input clusterings) over V that minimizes the objective function $\left(\sum_{i=1}^{m} (\mathit{dist}(\mathcal{D}_i, \mathcal{C}))^{\ell}\right)^{1/\ell}$.*

The generalized mean objective is well-studied in the clustering literature Chlamtác et al. (2022). In the context of consensus clustering, for $\ell = 1$, the above problem (note, the objective is now the sum of distances) is referred to as the *median consensus clustering* Ailon et al. (2008), and for $q = \infty$, the above problem (note, the objective is now the maximum distance) is the *center consensus clustering* Das and Kumar (2025).

Fair clustering.

Definition 8 (Fair clustering) *Consider a clustering $\mathcal{F} = \{F_1, F_2, \dots, F_\zeta\}$ on a set of red-blue colored points where the ratio between the number of blue and red points is p/q, for two integers $p, q \geq 1$. $\mathcal{F}$ is called a Fair clustering if for every cluster $F_i \in \mathcal{F}$, the ratio between the number of blue and red points $|blue(F_i)|/|red(F_i)| = p/q$.*

Next, we define the closest fair clustering.

Definition 9 (Closest Fair clustering) *Given a clustering $\mathcal{D} = \{D_1, D_2, \dots, D_\phi\}$, a clustering $\mathcal{F}^* = \{F_1^*, F_2^*, \dots, F_\iota^*\}$ is called its* closest Fair clustering *if for all Fair clustering $\mathcal{F}$, dist$(\mathcal{D}, \mathcal{F}^*) \leq$ dist$(\mathcal{D}, \mathcal{F})$.*

Let $\mathcal{F}^*$ be an (arbitrary) closest Fair clustering. Then, we use OPT^{fair}(or simply OPT when it is clear from the context) to denote *dist*$(\mathcal{D}, \mathcal{F}^*)$. For any $\alpha \geq 1$, a clustering $\mathcal{F}$ is called an *α-close* Fair clustering if *dist*$(\mathcal{D}, \mathcal{F}) \leq \alpha \cdot$ *dist*$(\mathcal{D}, \mathcal{F}^*)$.

In this paper, we focus on the *fair consensus clustering* problem.

Definition 10 (Generalized Mean Fair Consensus Clustering) *Consider an exponent parameter $\ell \in \mathbb{R}$. Given a set of m clusterings $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m$ over a point set V, the ℓ-*mean Fair Consensus Clustering *problem asks to find a Fair clustering $\mathcal{F}$ (not necessarily from the set of input clusterings) over V that minimizes the objective function $\left(\sum_{i=1}^{m} (dist(\mathcal{D}_i, \mathcal{F}))^\ell\right)^{1/\ell}$.*

Balanced clustering. One of the intermediate steps we use in making an input clustering fair is first to make it *balanced* and then convert a balanced clustering to a fair one.

Definition 11 (Balanced clustering) *Consider a clustering $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_\psi\}$ on a set of red-blue colored points where the irreducible[3] ratio between the number of blue and red points is p/q, for two integers $p, q \geq 1$. $\mathcal{Q}$ is called a Balanced clustering if for every cluster $Q_i \in \mathcal{Q}$, the number of blue points $|blue(Q_i)|$ is a multiple of p, and the number of red points $|red(Q_i)|$ is a multiple of q.*

Next, we define the closest balanced clustering.

Definition 12 (Closest Balanced clustering) *Given a clustering $\mathcal{D} = \{D_1, D_2, \dots, D_\phi\}$, a clustering $\mathcal{Q}^* = \{Q_1^*, Q_2^*, \dots Q_\tau^*\}$ is called its* closest Balanced clustering *if for all Balanced clustering $\mathcal{Q}$, dist$(\mathcal{D}, \mathcal{Q}^*) \leq$ dist$(\mathcal{D}, \mathcal{Q})$.*

Let $\mathcal{Q}^*$ be an (arbitrary) closest Balanced clustering. Then, we use OPT^{bal}(or simply OPT when it is clear from the context) to denote *dist*$(\mathcal{D}, \mathcal{Q}^*)$. For any $\alpha \geq 1$, a clustering $\mathcal{Q}$ is called an *α-close* Balanced clustering if *dist*$(\mathcal{D}, \mathcal{Q}) \leq \alpha \cdot$ *dist*$(\mathcal{D}, \mathcal{Q}^*)$.

[3] Note, a fraction p/q is called *irreducible* if p and q are coprime, i.e., $\mathsf{gcd}(p, q) = 1$.

3. Technical Overview

Given a set V of n points, a clustering $\mathcal{D} = \{D_1, D_2, \ldots, D_\phi\}$ is a partition of V into disjoint subsets. Given two clusters, we define their distance as the total number of pairwise disagreements between them. In this work, we assume that each point in V is associated with a color from the set $\{\text{red, blue}\}$. For a cluster D_i, let *blue*(D_i) denote the blue points in D_i, and *red*(D_i) denote the red points in D_i. Given a clustering $\mathcal{D}$, with the ratio between the number of blue and red points being p/q for some integer $p, q \geq 1$, we call $\mathcal{D}$ a *fair clustering* if, for each $i \in [\phi]$, the ratio between $|\mathit{blue}(D_i)|$ and $|\mathit{red}(D_i)|$ is exactly p/q. Next, we discuss how efficiently we can transform a given clustering $\mathcal{D}$ into its nearest fair clustering under various settings of p and q.

Optimal Fair Clustering for Equi-Proportionally Balanced Input. We begin by discussing the case where $p = q = 1$ and present a near-linear time algorithm that, given a clustering $\mathcal{D}$ of V, efficiently finds a closest fair clustering $\mathcal{C}^*$ (i.e., an optimal fair clustering). Below, we provide a brief outline of this algorithm.

Before describing the algorithm, we first establish key properties of the optimal fair clustering. In particular, we show that there exists an optimal fair clustering that satisfies specific structural properties. These properties serve as a guide for transforming $\mathcal{D}$ into $\mathcal{C}^*$.

(i) Our first observation considers an arbitrary cluster $D_i \in \mathcal{D}$. Without loss of generality, assume $|\mathit{blue}(D_i)| \geq |\mathit{red}(D_i)|$. We define the *maximal fair cluster* of D_i, denoted by $D_i^{\text{max,fair}}$, as the largest subset of D_i that contains exactly $|\mathit{red}(D_i)|$ red and blue points. Our first claim is that for each $i \in [\phi]$, the cluster $D_i^{\text{max,fair}}$ appears as a cluster in $\mathcal{C}^*$. The intuition behind this is as follows: it is always beneficial to retain at least $|\mathit{red}(D_i)|$ red and blue points from D_i together. Otherwise, if they are distributed across multiple clusters, the overall cost would increase, as these points originate from the same input cluster. Furthermore, if $D_i^{\text{max,fair}}$ were instead included as a subset of a larger cluster in $\mathcal{C}^*$, then since $\mathcal{C}^*$ is fair—the larger cluster would need additional blue points from another cluster to maintain fairness. Importantly, the number of foreign blue points added would be equal to the excess blue points removed when forming the maximal fair cluster, ensuring that the overall cost remains unchanged.

(ii) Next, consider a cluster $C_j^* \in \mathcal{C}^*$. We claim that C_j^* originates from at most two clusters in the input clustering $\mathcal{D}$. To see why this holds, assume for contradiction that C_j^* is formed from three distinct clusters $D_o, D_s, D_t \in \mathcal{D}$. From our earlier observations, each of these clusters can contribute only blue or only red points to C_j^*. Without loss of generality, suppose D_o contributes b_o blue points, while D_s and D_t contribute r_s and r_t red points, respectively. Since $\mathcal{C}^*$ is fair, we must have $b_o = r_s + r_t$. Now, assume without loss of generality that $r_s \leq r_t$. We can create a new clustering by partitioning C_j^* into two clusters: The first cluster contains r_s red points from D_s and r_s blue points from D_o. The second cluster contains r_t red points from D_t and r_t blue points from D_o. In this transformation, partitioning D_o increases the cost by $r_s \cdot r_t$. However, since the red points from D_s and D_t are now separated, the cost decreases by the same amount, $r_s \cdot r_t$. Thus, the overall cost remains unchanged. Moreover, by construction, the new clustering remains fair. In our algorithm, we extend this argument to a more general setting where C_j^* originates from more than three clusters, showing that a similar reasoning applies to establish the claim.

Using these two properties, we construct $\mathcal{C}^*$ from $\mathcal{D}$ through a two-step process.

In **Step 1:** Following observation (i), for each input cluster D_i, we extract the maximal fair cluster $D_i^{\text{max,fair}}$. Define the remaining points as $R_i = D_i \setminus D_i^{\text{max,fair}}$. Note that each R_i forms a monochromatic cluster.

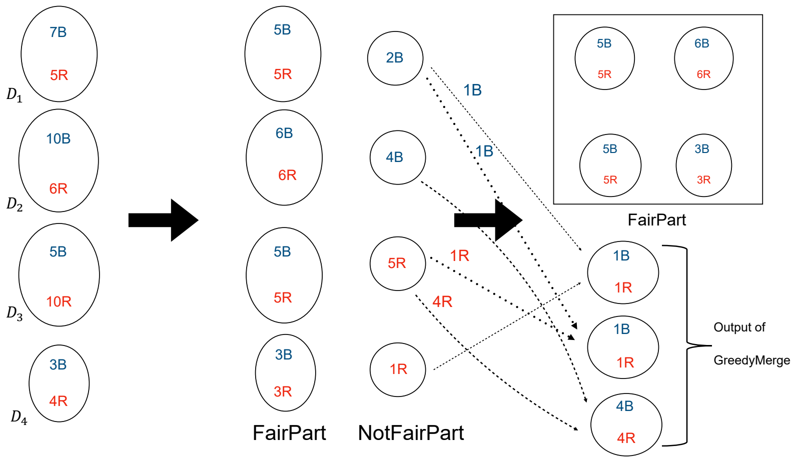

Figure 1: Visualization of the algorithm to find a closest fair clustering for equi-proportionally balanced input. In each cluster, xB (resp., yR) denotes that the number of blue (resp., red) points is x (resp., y).

In **Step 2**, we introduce a greedy strategy to combine $R_1, R_2, \ldots, R_\phi$ into a set of fair clusters. We begin by selecting the smallest cluster R_i. Without loss of generality, assume R_i consists of blue points. We then identify the smallest red cluster R_j and form a new cluster by combining R_i with $|R_i|$ red points from R_j. This process is repeated iteratively for the remaining monochromatic clusters. (See Fig. 1.)

By construction, this merging strategy preserves the fairness property. Additionally, since $\mathcal{C}^*$ is fair and each $D_i^{\text{max,fair}}$ is included in $\mathcal{C}^*$, the total number of red points in $\bigcup_{i\in[\phi]} R_i$ matches the total number of blue points. Thus, our greedy strategy ensures a fair partition of all remaining points.

Next, we show the optimality of this greedy strategy by leveraging Property (ii) of the optimal fair cluster discussed earlier. We claim that selecting the smallest R_i at each step is indeed optimal. Intuitively, it is preferable to keep all points of R_i together when they originate from the same input cluster. Further following Property (ii), the optimal fair cluster can contain points from at most two input clusters. Now, to make R_i fair while keeping it intact, we must merge it with a cluster R_j of the opposite color such that $|R_j| \geq |R_i|$. This condition is best satisfied by processing clusters in *non-decreasing* order of size, which justifies the rationale behind our greedy strategy.

Constant Approximation for Closest Fair Clustering (General p, q). We now consider a more general version of the problem for arbitrary values of p and q. Without loss of generality, we assume $p \geq q$. The algorithm proceeds in two steps:

1. **Balancing Step:** Given an input clustering $\mathcal{D} = \{D_1, D_2, \ldots, D_\phi\}$, we transform it into a new clustering $\mathcal{T} = \{T_1, T_2, \ldots, T_\gamma\}$ such that each cluster T_i contains a number of blue points that is a multiple of p and a number of red points that is a multiple of q.

2. **Making-Fair Step**: We further process the clusters $T_1, T_2, \ldots, T_\gamma$ to obtain a fair clustering $\mathcal{F} = \{F_1, F_2, \ldots, F_\zeta\}$.

Let $\mathcal{F}^*$ denote an optimal fair clustering closest to the input clustering $\mathcal{D}$, with an associated distance of OPT. By definition, $\mathcal{F}^*$ is also balanced.

If we propose an α-approximation algorithm for the balancing step, then the distance between $\mathcal{D}$ and $\mathcal{T}$ is at most $\alpha \cdot$ OPT. Furthermore, since the distance between $\mathcal{F}^*$ and $\mathcal{T}$ is at most $(1+\alpha) \cdot$ OPT (by the triangle inequality), getting a β-approximation algorithm for the making-fair step ensures that the distance between $\mathcal{T}$ and $\mathcal{F}$ is at most $(1+\alpha)\beta \cdot$ OPT.

Thus, the total distance between the input clustering $\mathcal{D}$ and the final output fair clustering $\mathcal{F}$ is at most $(\alpha + \beta + \alpha\beta) \cdot$ OPT, leading to an overall approximation ratio of $(\alpha + \beta + \alpha\beta)$. This guarantees a constant-factor approximation when α and β are constants.

3.5**-approximation Balancing Algorithm for** $p : 1$**:** We begin by analyzing the balancing step for the case where $q = 1$. For this, we present a 3.5-approximation algorithm.

For every cluster D_i, we define the *surplus* of D_i as $(|\mathit{blue}(D_i)| \mod p)$ and its deficit as $(p - (|\mathit{blue}(D_i)| \mod p))$. To balance each cluster, we either need to remove surplus points or add the required number of deficit points from another cluster. For each cluster, we define the *cut cost* as the cost of removing surplus points and the *merge cost* as the cost of adding deficit points.

Next, we define our cut and merge strategy. We start by classifying clusters into two categories. Call a cluster *merge cluster* if its surplus is at least $p/2$; i.e., its merge cost is smaller than its cut cost. Otherwise, call it a *cut cluster*. For a merge cluster D_i, it is more efficient to add $(p - (|\mathit{blue}(D_i)| \mod p))$ points (its deficit), whereas, for a cut cluster D_i, it is optimal to remove $(|\mathit{blue}(D_i)| \mod p)$ points (its surplus). A first-line approach is to remove surplus points from cut clusters and merge them into merge clusters, ensuring that the number of blue points in each cluster becomes a multiple of p. We start with this approach, referred to as *AlgoGeneral*. However, we cannot guarantee that the total surplus across all clusters matches the total deficit. As a result, after this step, we may be left with only cut clusters or only merge clusters.

If only merge clusters remain, we cannot satisfy all merge requests since we need to cut some clusters to provide the necessary points. This is challenging, as cutting is now costlier than merging for the remaining unbalanced clusters. The key intuition here is that if cutting is necessary, we need a nontrivial strategy to optimize the cost. To achieve this, we sort the merge clusters in non-increasing order of $cutcost - mergecost$, ensuring optimal cutting decisions. An important observation is that the total deficit W across all clusters is a multiple of p. This follows from the existence of a fair clustering, which guarantees that there exists a way to process the input set of clusters such that the number of blue points in each cluster is a multiple of p. Additionally, if we cut k points from a cluster D_i to fulfill the k deficit of another cluster D_j, we effectively balance a total deficit of p points. This follows intuitively because if k is the surplus of D_i, then its deficit is $p-k$, ensuring that the combined deficit of D_i and D_j sums up to p. Since the total deficit is W, we repeat this operation exactly W/p times, which is crucial for bounding the overall approximation cost. (See Fig. 2.)

Next, we consider the simpler case where, after the initial cut-merge processing, we are left only with cut clusters, each having a surplus of size $< p/2$. Here, we simply remove these surplus points from each cluster and combine them to form clusters of size exactly p, containing only blue points. A key question is why forming clusters of size p is the optimal choice rather than larger clusters. The reason is that ensuring size p may require partitioning surplus points from specific clusters, incurring additional costs. For example, consider the case where there are three cut clusters with surpluses of $p/3$, $p/3$, and $p/2 - 1$. Following our strategy to form a cluster of size p, we must partition at least one existing cluster. One might wonder whether it is more efficient to combine

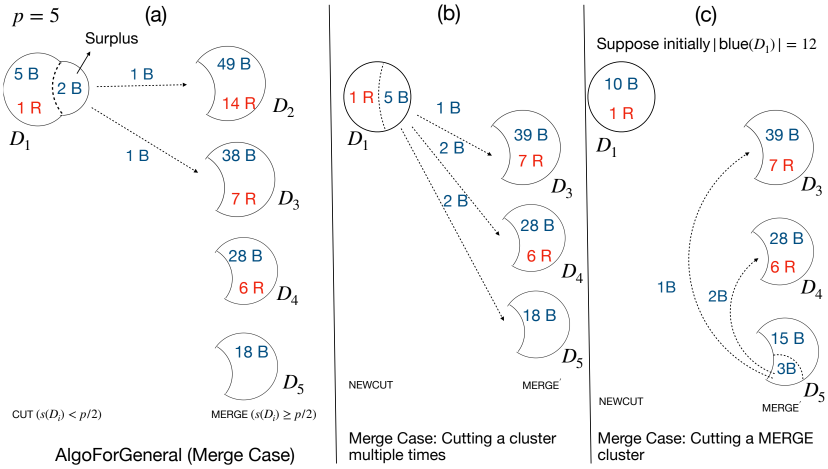

Figure 2: Visualization of the scenario when only merge clusters remain after executing AlgoGeneral, which converts (a) to (b). Now, the total deficit $W = 5$, and thus, depending on the $cutcost - mergecost$, either an already cut cluster needs to be cut further (depicted in (b)), or a single merge cluster needs to be cut (depicted in (c)). In each cluster, xB (resp., yR) denotes that the number of blue (resp., red) points is x (resp., y).

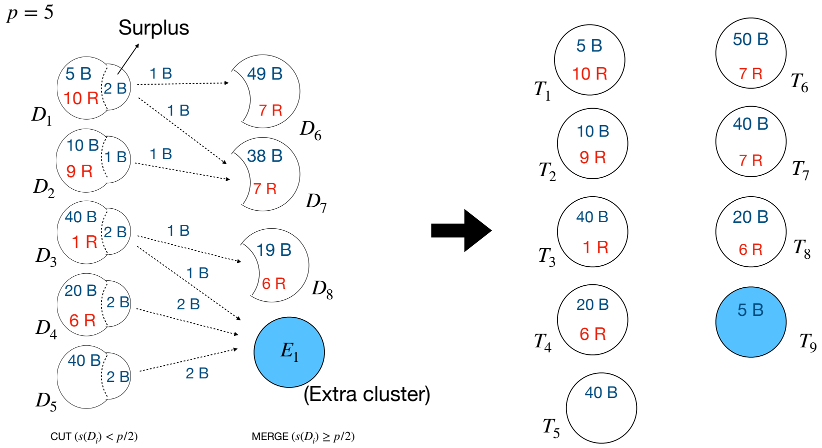

Figure 3: Visualization of the scenario when only cut clusters remain after executing AlgoGeneral. The extra surplus points of clusters D_3, D_4, D_5 form an extra cluster E_1 of size p, containing only blue points. In each cluster, xB (resp., yR) denotes that the number of blue (resp., red) points is x (resp., y).

even more clusters and create a cluster of size $2p$ instead of p. However, through careful analysis, we show that accommodating clusters of size $> p$ results in higher merge costs than the partition cost. Thus, forming clusters of size exactly p is the optimal strategy for combining surplus points efficiently. (See Fig. 3.)

7.5-approximation Balancing Algorithm for $p : q$: We now turn our attention to balancing when both p and q are greater than 1. We build on the fundamental concept of the balancing step previously outlined for the $p : 1$ scenario. In this context, we aim to ensure that the blue subpart of each cluster is a multiple of p and the red subpart is a multiple of q. This involves examining surpluses and deficits for both the blue and red subparts separately, similar to the $p : 1$ case. Consequently, each cluster D_i can be categorized into one of four types: blue merge - red cut, blue cut - red merge, blue cut - red cut, and blue merge - red merge, where blue (resp., red) merge denotes a deficit of at most $p/2$ points (resp., $q/2$ points) and blue (resp., red) cut denotes a surplus of at most $p/2$ points (resp., $q/2$ points). Depending on its type, we employ our previously outlined balancing algorithm by addressing the blue and red components of a cluster D_i separately.

However, with two colors, our analysis now encounters additional challenges in ensuring an approximation bound. For example, consider a cluster D_i of type blue merge - red merge. An optimal strategy might involve merging external blue points (denoted as B_i) to address the blue deficit and external red points (denoted as R_i) to address the red deficit. However, this introduces an intercluster cost between the sets B_i and R_i, calculated as $|R_i| \cdot |B_i|$. In this context, it can be noted that $|R_i| \leq q/2 \leq |\mathit{red}(D_i)|$, which means the additional cost is bounded above by OPT_{D_i}, where OPT_{D_i} represents the optimal cost incurred by the points within cluster D_i. To understand this, note that OPT_{D_i} must cover at least the cost of resolving its blue deficit, which is at least $|\mathit{red}(D_i)| \cdot |B_i|$. Summing these costs across all such clusters results in an additional cost that is bounded above by the overall optimal cost.

For the remaining three cases, we show that by adapting the techniques and arguments from the $p : 1$ case, we achieve the claimed approximation guarantee. Finally, by summing all these costs, we establish that the overall approximation factor is bounded by 7.5.

Making the Balance Clustering Fair: Given a balanced clustering $\mathcal{T} = \{T_1, T_2, \ldots, T_\gamma\}$ here we discuss the step to convert it to a fair clustering $\mathcal{F} = \{F_1, F_2, \ldots, F_\zeta\}$. For this, we present a 3-approximation algorithm.

Without loss of generality, assume $p \geq q$. We define a cluster T_i as TYPERED if it has surplus red points, meaning $|\mathit{blue}(T_i)| < (p/q)|\mathit{red}(T_i)|$. Let S_i denote the surplus of red points in T_i, given by $|S_i| = |\mathit{red}(T_i)| - (q/p)|\mathit{blue}(T_i)|$. Otherwise, we define T_i as TYPEBLUE if it has a deficit of red points, meaning $|\mathit{blue}(T_i)| > (p/q)|\mathit{red}(T_i)|$. Let D_i represent the red point deficit in T_i, given by $|D_i| = (q/p)|\mathit{blue}(T_i)| - |\mathit{red}(T_i)|$.

Next, we discuss how to eliminate surplus red points and redistribute them to address the deficits. To achieve this, we simply remove the surplus points from clusters in TYPERED and reassign them to clusters in TYPEBLUE to compensate for the deficit. A crucial observation is that the existence of a fair clustering guarantees that the total surplus from all clusters in TYPERED matches the total deficit across all clusters in TYPEBLUE, thereby making the redistribution process well-defined. We now proceed to analyze the approximation.

We first claim that since $p \geq q$, removing surplus points and redistributing them to fulfill deficits is indeed optimal. A key subtlety we need to address is the case where a specific surplus S_i is distributed across multiple clusters to fulfill their deficits. This redistribution involves splitting S_i, which incurs an additional cost. However, even in an optimal fair clustering, S_i must be transformed into a fair cluster, requiring the merging of $(p/q)|S_i|$ blue points with S_i. The cost of this merging is $(p/q)|S_i|^2$, while the maximum splitting cost of S_i is at most $|S_i|^2/2$. Since $p \geq q$, the splitting cost of S_i remains bounded by its merging cost, ensuring the overall 3-approximation guarantee. In the case where multiple surplus clusters contribute to fulfilling a deficit D_i, our algorithm incurs an intra-cluster cost of at most $|D_i|^2/2$. Since $p \geq q$, a similar argument as before ensures that, despite this additional cost, we still remain within the claimed approximation guarantee.

Fair Consensus Clustering. Next, we propose an algorithm for *Fair Consensus Clustering* that minimizes the ℓ-mean objective for any $\ell \geq 1$. Formally, given m clusterings $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_m$ on a set V, the goal is to find a fair clustering $\mathcal{F}^*$ that minimizes $\left(\sum_{j=1}^{m} (\mathit{dist}(\mathcal{D}_j, \mathcal{F}^*))^\ell\right)^{1/\ell}$. To achieve this, we design a simple algorithm leveraging our *Closest Fair Clustering* algorithm discussed above.

We begin by computing, for each $\mathcal{D}_i$, an α-close fair clustering $\mathcal{F}_i$ using the *Closest Fair Clustering algorithm*. Then, we output the fair clustering $\mathcal{F}_k$ that minimizes the overall ℓ-mean objective. Next, we show that this approach guarantees a $(2+\alpha)$-approximation for *Fair Consensus Clustering*.

Let $\mathcal{F}^*$ be an optimal fair consensus clustering, and let $\mathcal{D}_i$ be the input clustering closest to $\mathcal{F}^*$. Using our α-approximate *Closest Fair Clustering* algorithm, we obtain a fair clustering $\mathcal{F}_i$ such that $\mathit{dist}(\mathcal{D}_i, \mathcal{F}_i) \leq \alpha \mathit{dist}(\mathcal{D}_i, \mathcal{F}^*)$. By applying the triangle inequality, we get $\mathit{dist}(\mathcal{F}_i, \mathcal{F}^*) \leq (1+\alpha)\mathit{dist}(\mathcal{D}_i, \mathcal{F}^*)$. Now, for any other input clustering $\mathcal{D}_j$, we have

$$\mathit{dist}(\mathcal{D}_j, \mathcal{F}_i) \leq \mathit{dist}(\mathcal{D}_j, \mathcal{F}^*) + \mathit{dist}(\mathcal{F}_i, \mathcal{F}^*) \leq (2+\alpha)\mathit{dist}(\mathcal{D}_j, \mathcal{F}^*),$$

since by assumption, $\mathcal{D}_i$ is the input closest to $\mathcal{F}^*$, ensuring $\mathit{dist}(\mathcal{D}_i, \mathcal{F}^*) \leq \mathit{dist}(\mathcal{D}_j, \mathcal{F}^*)$.

Thus, we conclude that there exists an input clustering $\mathcal{D}_i$ whose corresponding fair clustering $\mathcal{F}_i$ achieves a $(2+\alpha)$-approximation. Since our algorithm selects the fair clustering $\mathcal{F}_k$ that minimizes the overall objective, it guarantees an overall approximation of $(2+\alpha)$.

Moreover, as our proposed *Closest Fair Clustering* algorithm achieves a constant approximation factor α, this ensures that our *Fair Consensus Clustering* algorithm also provides a constant-factor approximation.

All the details of the algorithms and analysis are provided in the full version[4].

4. Conclusion

In this paper, we initiate the study of closest fair clustering and fair consensus clustering problems. We focus on datasets where each point is associated with a binary protected attribute (red or blue). First, we study the problem of transforming an arbitrary clustering into a fair clustering while minimizing modifications. For perfectly balanced datasets with an equal number of red and blue points, we propose an optimal algorithm. We then extend our approach to more general cases with arbitrary ratios and develop constant-factor approximation algorithms. Building on our closest fair clustering algorithm, we develop a constant-factor approximation algorithm for Fair Consensus Clustering.

A natural open question is whether the approximation factors for both problems can be improved. For the closest fair clustering problem, directly achieving a better approximation is of particular interest. In contrast, the fair consensus clustering problem is effectively solved through a reduction to the closest fair clustering, making it compelling to explore whether a tighter reduction can be achieved in terms of the approximation guarantee. Another promising direction is extending

[4] The full version of this paper is available at https://arxiv.org/abs/2506.08673.

these methods to handle non-binary and potentially overlapping protected groups, broadening the applicability of fair clustering frameworks.

Acknowledgments

Diptarka Chakraborty was supported in part by an MoE AcRF Tier 2 grant (MOE-T2EP20221-0009), an MoE AcRF Tier 1 grant (T1 251RES2303), and a Google South & South-East Asia Research Award. Kushagra Chatterjee was supported by an MoE AcRF Tier 2 grant (MOE-T2EP20221-0009). Debarati Das was supported in part by NSF grant 2337832. A part of the research was done when Romina Nobahari was doing an internship at the National University of Singapore, where she was supported by an MoE AcRF Tier 2 grant (MOE-T2EP20221-0009).

References

Saba Ahmadi, Sainyam Galhotra, Barna Saha, and Roy Schwartz. Fair correlation clustering. *arXiv preprint arXiv:2002.03508*, 2020.

Sara Ahmadian and Maryam Negahbani. Improved approximation for fair correlation clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 9499–9516. PMLR, 2023.

Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. Fair correlation clustering. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108, pages 4195–4205, 2020.

Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):1–27, 2008.

Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. Scalable fair clustering. In *International Conference on Machine Learning (ICML)*, volume 97, pages 405–413, 2019.

Suman Kalyan Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4955–4966, 2019.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016.

Paola Bonizzoni, Gianluca Della Vedova, Riccardo Dondi, and Tao Jiang. On the approximation of correlation clustering and consensus clustering. *J. Comput. Syst. Sci.*, 74(5):671–696, 2008.

L Elisa Celis, Damian Straszak, and Nisheeth K Vishnoi. Ranking with fairness constraints. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 107, pages 28:1–28:15, 2018.

Diptarka Chakraborty, Syamantak Das, Arindam Khan, and Aditya Subramanian. Fair rank aggregation. *Advances in Neural Information Processing Systems*, 35:23965–23978, 2022. Full version: arXiv preprint arXiv:2308.10499.

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

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

Eden Chlamtác, Yury Makarychev, and Ali Vakilian. Approximating fair clustering with cascaded norm objectives. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *ACM-SIAM Symposium on Discrete Algorithms, (SODA)*, pages 2664–2683, 2022.

Debarati Das and Amit Kumar. Breaking the two approximation barrier for various consensus clustering problems. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 323–372. SIAM, 2025.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science*, pages 214–226, 2012.

Seyed Esmaeili, Brian Brubach, Leonidas Tsepenekas, and John Dickerson. Probabilistic fair clustering. *Advances in Neural Information Processing Systems*, 33:12743–12755, 2020.

Vladimir Filkov and Steven Skiena. Heterogeneous data integration with the consensus clustering formalism. In *International Workshop on Data Integration in the Life Sciences*, pages 110–123. Springer, 2004a.

Vladimir Filkov and Steven Skiena. Integrating microarray data by consensus clustering. *International Journal on Artificial Intelligence Tools*, 13(04):863–880, 2004b.

Andrey Goder and Vladimir Filkov. Consensus clustering algorithms: Comparison and refinement. In *2008 Proceedings of the Tenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 109–117. SIAM, 2008.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3315–3323, 2016.

Lingxiao Huang, Shaofeng H.-C. Jiang, and Nisheeth K. Vishnoi. Coresets for clustering with fairness constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7587–7598, 2019.

Matthew Kay, Cynthia Matuszek, and Sean A Munson. Unequal representation and gender stereotypes in image search results for occupations. In *ACM conference on human factors in computing systems*, pages 3819–3828, 2015.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. Human decisions and machine predictions. *The quarterly journal of economics*, 133(1):237–293, 2018.

Andrii Kliachkin, Eleni Psaroudaki, Jakub Mareček, and Dimitris Fotakis. Fairness in ranking: Robustness through randomization without the protected attribute. In *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*, pages 201–208. IEEE, 2024.

Mirko Křivánek and Jaroslav Morávek. Np-hard problems in hierarchical-tree clustering. *Acta informatica*, 23:311–323, 1986.

Andrea Lancichinetti and Santo Fortunato. Consensus clustering in complex networks. *Scientific reports*, 2(1):336, 2012.

Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, 52:91–118, 2003.

Chaitanya Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *SODA*, volume 4, pages 526–527. Citeseer, 2004.

Alexander Topchy, Anil K Jain, and William Punch. Combining multiple weak clusterings. In *Third IEEE international conference on data mining*, pages 331–338. IEEE, 2003.

Dong Wei, Md Mouinul Islam, Baruch Schieber, and Senjuti Basu Roy. Rank aggregation with proportionate fairness. In *SIGMOD International Conference on Management of Data*, pages 262–275, 2022.

Junjie Wu, Hongfu Liu, Hui Xiong, Jie Cao, and Jian Chen. K-means-based consensus clustering: A unified view. *IEEE transactions on knowledge and data engineering*, 27(1):155–169, 2014.