The Fairness-Quality Trade-off in Clustering

Rashida Hakim
Columbia University

Ana-Andreea Stoica
Max Planck Institute for
Intelligent Systems, Tübingen

Christos H. Papadimitriou
Columbia University

Mihalis Yannakakis
Columbia University

Abstract

Fairness in clustering has been considered extensively in the past; however, the trade-off between the two objectives — e.g., can we sacrifice just a little in the quality of the clustering to significantly increase fairness, or vice-versa? — has rarely been addressed. We introduce novel algorithms for tracing the complete trade-off curve, or Pareto front, between quality and fairness in clustering problems; that is, computing all clusterings that are not dominated in both objectives by other clusterings. Unlike previous work that deals with specific objectives for quality and fairness, we deal with all objectives for fairness and quality in two general classes encompassing most of the special cases addressed in previous work. Our algorithm must take exponential time in the worst case as the Pareto front itself can be exponential. Even when the Pareto front is polynomial, our algorithm may take exponential time, and we prove that this is inevitable unless P = NP. However, we also present a new polynomial-time algorithm for computing the entire Pareto front when the cluster centers are fixed, and for a fairness objective that minimizes the sum, over all clusters, of the imbalance between the two groups in each cluster.

1 Introduction

Clustering is a fundamental problem in unsupervised learning with many applications, in which data points must be grouped into clusters so that the *quality* or *cost* of the clustering is optimized — where the cost can be k-median or k-center, among a long array of different quality objectives treated in the literature. In some of these applications, the data points being clustered are people — for example, when households are clustered to determine the location of a bus stop or a hospital — and in those cases considerations such as fairness and equity come into play. In recent work, various frameworks and methods have been proposed for improving the representation of different sensitive groups in clustering, in cases where optimizing for clustering cost alone may be quite unfair. In such works, the fairness criterion is often set as a constraint, for which we must compute the optimal solution. This is generally an intractable computational problem even for the simplest fairness objectives, since clustering on its own is NP-hard. The best one could hope for is an approximate solution based on a vanilla approximation algorithm for clustering that improves fairness; such approaches have indeed been recently proposed, e.g., Esmaeili et al. [26], see Chhabra et al. [18] for a survey.

We take a more general approach to fair clustering: We aim to recover the entire *Pareto front* of clustering cost and fairness, rather than a single point of that front (such as the best-quality clustering under a fairness constraint). Our work aims to enable a practitioner to choose any trade-off point, rather than computing one of them. We give general sufficient conditions for which we can recover the Pareto front, up to an approximation inherited from the clustering problem itself, and encompassing a variety of quality and fairness objectives used in the literature.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

1.1 Our Contributions

We present a novel family of algorithms for tracing the Pareto front between a quality/cost objective and a fairness objective, where each objective can belong in a general class that encompasses most of the ones previously proposed in the literature. Our algorithm requires that the fairness objective satisfy an intuitive property: that it is *pattern-based*. Informally, a fairness objective is pattern-based if it is solely a function of the number of nodes of different attributes in each cluster (see Definition 2.1). Many fairness objectives used in the literature are pattern-based, from the well-known balance [19] to objectives that measure the proportional violation of pre-specified upper and lower bounds for a sensitive attribute within a cluster [10, 40, 2, 9, 26, 32, 22]. For the quality objective, we consider metric-based cost functions. We study the computation of the quality/fairness trade-off in two settings: in the *assignment problem*, the centers of the clusters are given and the cost is the sum of the distances of the data points from their cluster center; whereas in the *clustering problem*, determining the centers is a part of the problem. We show that our algorithm finds the exact Pareto front of the assignment problem, and an approximation of the Pareto front of the clustering problem (Theorems 3.4, 3.5), with an approximation ratio that depends on the baseline approximation ratio (denoted α) for the minimum cost clustering problem without a fairness objective. The running time is $O(kn^{l(k-1)})$, where k is the number of clusters, l is the number of sensitive attribute values, and n is the size of the dataset.

Our Pareto front algorithms take time exponential in ℓ and k, and this is necessary. One reason is that the Pareto front may be exponential in size. For many fairness objectives the Pareto front is provably polynomial in size, but even in those cases exponential time is still necessary in general because the baseline problem is NP-hard. Importantly, we also present the first nontrivial algorithmic result for the quality/fairness Pareto front problem in clustering: A *polynomial-time algorithm* for computing the Pareto front for a new fairness objective, namely the one minimizing the sum (or the maximum), over all clusters, of the deviations from parity of the two protected attributes in each cluster (see Theorem 3.6).

We empirically explore faster methods for computing an approximate Pareto front by adapting constraint optimization methods from fair clustering [26] in Section 4. These experiments explore the trade-off between accuracy and speed for our problem by comparing the tighter approximation obtained through our proposed algorithms with the looser Pareto front approximation obtained using faster methods.

We believe that our work is the first to address and carry out the computation of the entire Pareto front, and also the first to simultaneously cover a large range of quality and fairness criteria. In comparison, extant literature on fair clustering typically optimizes one objective subject to a constraint on the other objective, with an individualized approach for each combination. Our work is particularly applicable in cases where a decision-maker may be willing to be suboptimal in one objective in order to achieve a better value in a second objective, but does not know *a priori* what bounds to set on either objective. Our algorithms allow them to explore the entire trade-off, from the highest quality clustering all the way down to the fairest clustering. Finally, our polynomial-time algorithm for computing the Pareto front when the fairness objective is the sum or maximum of imbalances (see Theorem 3.6) is another novel contribution to the subject of fair clustering.

1.2 Related Work

Our work is inspired by a plethora of recent studies on fair clustering that propose various metrics for improving the representation of different sensitive attributes within clusters. Unsupervised clustering is known to be NP-hard even in simple settings [4]. A variety of algorithms have been proposed to approximate the best possible clustering, with approximation ratios differing for different cost objectives (e.g., k-means [3], k-median [14], k-center [38]). Fair clustering goals range from maximizing the worst ratio between two groups across clusters [19], to minimizing the discrepancy between cluster representation and proportional representation of a group in a population [2, 9, 10, 22, 32, 26, 40], and to equalizing the clustering cost for various groups [7, 15, 31, 50, 54, 62]. Many of these works focus on building specialized approaches for each of the fairness objectives defined, and their objective is to find the best quality clustering that satisfies a specified fairness constraint. Optimizing even for simplest fairness objectives can be NP-hard [26], with approximation guarantees involving a multiplicative and additive factor, which may depend on the particular objective form and data topology. Our work differs in two significant ways: first, we compute the entire trade-off curve rather than a single optimization point on the curve; second, we provide algorithms that are agnostic to the specific objectives used, giving sufficient conditions on the objectives. We note that our conditions favor most group-fairness objectives defined in the literature. We discuss extensions to other definitions of fairness and limitations of our work in Section 5.

Our work takes a different approach, by aiming to compute the entire Pareto front between seemingly opposing objectives. Knowing the entire Pareto front can aid decision-makers when faced with choosing optimal trade-offs. Many real-world applications are faced with such trade-offs: for example, in facility location problems where a central agent decides on the location of facilities (e.g., buses, hospitals, etc.) based on the location of individuals within a certain region. It is well-known that people of lower socio-economic status often travel longer distances [57] and have fewer health facilities in their region [29]. A decision maker has multiple objectives to balance: minimal travel time for the entire population and equal access to facilities for different populations. The Pareto front allows the decision maker to balance these objectives in the optimal way: how much improvement can a group of people gain by moving a facility slightly away from the solution given by a single objective (e.g., by performing k-means)? Computing the Pareto front for facility location problems with multiple objectives has a long history motivated by such questions, with real-world case studies in the Singapore road network [39] and ambulance placement in Barbados [35]. Further applications are found in extensive surveys [28].

From a theoretical standpoint, computing the entire Pareto front remains a difficult problem. When an underlying single-objective optimization problem is NP-hard, such as in many combinatorial optimization problems, computing an approximation for the Pareto front is the best one could hope for [55], for example for the multi-objective shortest path problem [11, 36, 44, 64], zero one knapsack problem [25, 42, 63], the multi-objective spanning tree problem [8, 21, 55], among others. To our knowledge, our work is novel in proposing algorithms for solving a bi-objective optimization problem based on clustering and fairness objectives which require only general properties for the fairness objectives to recover an approximate Pareto front with theoretical guarantees.

More generally, multi-objective optimization has seen a variety of methods for computing points close to the Pareto front in various domains, with methods ranging from evolutionary algorithms [20, 56, 61, 65] to gradient-based methods [12, 47, 49, 51], and more recently to hypernetworks [16, 37, 48, 60] (often requiring inputting a preference vector of the objective values that will output a point on the Pareto front). These methods have also been applied to a problem closely related to clustering: that of facility location with multiple objectives [34, 58, 59], considering specific objectives and without theoretical approximation guarantees.

2 Preliminaries

We denote by $\mathcal{X}$ the set of data points in $\mathbb{R}^d$. Assume that $\mathcal{X} = \{x_1, \ldots, x_n\}$, and for $j \leq n$ define $\mathcal{X}_j = \{x_1, \ldots, x_j\}$. A *clustering map* is defined as $\phi : \mathcal{X} \to S$, where S is a set of k cluster centers in $\mathbb{R}^d$ and k is the number of clusters, considered fixed. A cluster C_i is defined as all the data points for which $\phi(x) = s_i$, where $s_i \in S$ is the center for the i-th cluster. We call a *clustering* $\mathcal{C}$ the set of all clusters C_i (so $|\mathcal{C}| = k$), and by $\mathcal{K}$ the set of all possible clusterings. Finally, we denote by $\sigma : \mathcal{X} \to [l]$ the map between data points and a set of sensitive attributes, indexed from 1 to l (which may represent demographic groups, interest groups, etc). We denote by C^a the set of data points in a cluster C with attribute a, and by $\mathcal{X}^a$ the set of data points with attribute a.

Clustering and Assignment Problems. Unsupervised clustering optimizes a clustering cost objective c, often defined as the sum of distances between points and a set of candidate cluster centers. In a general form, the cost of a clustering is $\left(\sum_{x \in \mathcal{X}} d^p(x, \phi(x))\right)^{1/p}$, for metric d and some value of p. We call such objectives *metric-based*. By varying $p = 1, 2, \infty$ we can obtain the k-median, k-means, and k-center objectives, respectively. The *clustering problem* is thus finding the clustering that has the minimum cost, over all possible sets of centers and maps from $\mathcal{X}$ to the set of centers. Since the clustering problem is hard, one often considers, as a stepping stone, the *assignment problem*, where the centers have been fixed. We shall consider both problems in this paper. In the fair clustering problem we have two objectives, the clustering cost objective c, and a fairness objective f, that aims to balance the representation of sensitive attributes in clusters, further detailed below.

Pareto Front. Consider the set of all clusterings $\mathcal{K}$ of $\mathcal{X}$, and the two objectives c (the cost objective) and f (the fairness objective[1]), each mapping $\mathcal{K}$ to $\mathbb{R}$. We say that clustering $\mathcal{C}$ dominates clustering $\mathcal{C}'$ if $c(\mathcal{C}) \leq c(\mathcal{C}')$, $f(\mathcal{C}) \leq f(\mathcal{C}')$, and one of the inequalities is strict. Intuitively, if a clustering $\mathcal{C}'$ is dominated, it is unworthy of further consideration, because it lags behind in both objectives of interest. If however a clustering $\mathcal{C}$ is *un*dominated, that is, there is no clustering in $\mathcal{K}$ that is simultaneously better on both fronts, then it is part of the solution. The *Pareto front* is the set of all undominated clusterings.

Fairness Objectives. Our main contribution is an algorithm for computing the Pareto front of the clustering and assignment problems for any metric-based quality function, and any fairness objective – always a function to be minimized – that satisfies the following general condition.

Definition 2.1 (Pattern-based objectives). For a clustering $\mathcal{C} \in \mathcal{K}$, $i \in [k]$, and $a \in [l]$, let its *pattern* $P^{\mathcal{C}}[i, a]$ be the number of data points in cluster C_i with attribute value a. A fairness objective f that maps $\mathcal{K}$ to $\mathbb{R}$ is called *pattern-based* if $f(\mathcal{C})$ only depends on the values in $P^{\mathcal{C}}[i, a]$.

Definition 2.2 (Mergeable objectives). Consider two clusterings $\mathcal{C}$ and $\mathcal{C}'$. We say that $\mathcal{C}'$ is the result of a merging of $\mathcal{C}$ (or $\mathcal{C}$ is a refinement of $\mathcal{C}'$) if every non-empty cluster of $\mathcal{C}'$ is the union of clusters of $\mathcal{C}$. A fairness objective f that maps $\mathcal{K}$ to $\mathbb{R}$ is called *mergeable* if, whenever $\mathcal{C}'$ is the result of a merging of $\mathcal{C}$, $f(\mathcal{C}') \leq f(\mathcal{C})$.

We discuss the pattern-based and mergeability properties in Appendix B, where we give examples of fairness objectives that are not pattern-based or mergeable. Below, we introduce several fairness objectives commonly found in the literature, which we use in our experiments. All of these objectives are pattern-based and mergeable, as proved formally in Appendix B.1.

Balance objective (Definition 1 *in Chierichetti et al. [19]):* For sensitive attributes that can take two values, indexed 1 and 2, the balance of a cluster C is defined as $\textsc{Balance}(C) = \min\left(|C^1|/|C^2|, |C^2|/|C^1|\right)$. The balance objective for a clustering $\mathcal{C}$ is then defined as

$$\textsc{Balance}(\mathcal{C}) = \max \min_{C \in \mathcal{C}} \textsc{Balance}(C) \tag{1}$$

The aim is to maximize balance, or equivalently, to minimize the negative balance $f(\mathcal{C}) = -\textsc{Balance}(\mathcal{C})$. $\textsc{Balance}$ has been used in fair clustering as a measure of equalizing proportions of different groups across clusters [19]. In practice, optimizing $\textsc{Balance}$ is both difficult and does not measure how far the proportions of sensitive groups in a clustering are from their *true* proportions in the population. For this reason, objectives based on *proportional violation* have been proposed, allowing a central decision-maker to input upper and lower bounds for desired proportions of groups in each cluster, and measured the deviation from these bounds [2, 9, 10, 22, 26, 32, 40].

Proportional violation objectives: As defined in Esmaeili et al. [26], for every sensitive attribute $a \in [l]$, define upper and lower bounds α_a and β_a with the aim of satisfying $\beta_a|C| \leq |C^a| \leq \alpha_a|C|$ for all clusters $C \in \mathcal{C}$. Since this is not always feasible, we define the worst proportional violation of attribute a in cluster C as the minimum non-negative value $\Delta_a^C \in [0, 1]$ such that

$$\left(\beta_a - \Delta_a^C\right)|C| \leq |C^a| \leq \left(\alpha_a + \Delta_a^C\right)|C| \tag{2}$$

Then, the proportional violation-based objectives are defined as:

$$\textsc{Group Utilitarian} = \min \sum_{a \in [l]} \max_{C \in \mathcal{C}} \Delta_a^C, \qquad \textsc{Group Utilitarian-Sum} = \min \sum_{a \in [l], C \in \mathcal{C}} \Delta_a^C, \tag{3}$$

$$\textsc{Group Egalitarian} = \min \max_{a \in [l], C \in \mathcal{C}} \Delta_a^C, \qquad \textsc{Group Egalitarian-Sum} = \min \max_{a \in [l]} \sum_{C \in \mathcal{C}} \Delta_a^C \tag{4}$$

These objective operationalize utilitarian and egalitarian concepts from social choice [13], minimizing either the sum of proportional violations or the worst violation across attributes.

Sum of imbalances objective: Finally, for two interest groups in the population ($l = 2$) the following objective is quite natural:

$$\textsc{Sum of Imbalances} = \sum_{i \in [k]} |C_i^1 - C_i^2|,$$

[1] We will treat f as a minimization objective, thus minimizing the *un*fairness of a clustering.

that is to say, the sum of the deviations from equality between the two attribute values in the clusters. This objective is most appropriate when datasets contain relatively equal proportions of the two groups. Rather remarkably, the Pareto front for this objective can be computed in polynomial time.

3 Algorithms for Computing the Pareto Front

3.1 A Dynamic Programming Algorithm for Recovering the Assignment Pareto Front

We shall now present the main Algorithm. Define an $\mathcal{X}$-pattern P to be a k-tuple of l-tuples of non-negative integers such that $\sum_{i\in[k]} P[i,a] = |\mathcal{X}^a|, \forall a \in [l]$. $P[i,a]$ specifies the number of data points in cluster i of attribute a. That is, an $\mathcal{X}$-pattern is a clustering except only the attribute values of the points have been specified.[2] Complete proofs to all subsequent results can be found in Appendix A.

Algorithm 1 Dynamic Programming Algorithm for Computing the Assignment Pareto Front

```
Input: Number of clusters $k$, a set of $n$ data points $\mathcal{X}$ with $l$ attribute values, $k$ centers $S = \{s_1, \dots, s_k\}$, and metric-based cost objective $c$ parameterized by $d, p$.
Output: A table $T_n$ containing the solutions of the assignment problem for $\mathcal{X}$ for all $\mathcal{X}$-patterns.
Method: Dynamic programming. We shall compute $T_0, T_1, \dots T_n$.
Initialize $T_0$ to contain the null pattern with cost 0 and the empty clustering.
for $j = 1$ to $n$ do
    Generate all $\mathcal{X}_j$-patterns, where $\mathcal{X}_j = \{x_1, \dots, x_j\}$.
    For each $\mathcal{X}_j$-pattern $P$, and for each cluster $i$ such that $P[i,a] > 0$, where $a$ is the attribute
    value of $x_j$, look up in $T_{j-1}$ the cost of the pattern $P_i$, which is $P$ with $x_j$ omitted from
    cluster $i$, and compute $T_{j-1}(P_i) + d^p(x_j, s_i)$.
    Let $i^*$ be the cluster index that minimizes $T_{j-1}(P_i) + d^p(x_j, s_i)$.
    Set $T_j(P) \leftarrow T_{j-1}(P_{i^*}) + d^p(x_j, s_{i^*})$.
    Store at $T_j(P)$ the clustering from $T[P_{i^*}]$, with $x_j$ added at cluster $i^*$.
return $T_n$
```

At the conclusion of the algorithm, the table T_n contains the lowest cost clustering $\mathcal{C}$ for each $\mathcal{X}$-pattern P, such that $P^{\mathcal{C}} = P$, together with its cost $c(\mathcal{C})$. Then, we can find the Pareto front by first sorting these clusterings for all $\mathcal{X}$-patterns P in increasing c, and then traversing them in order, computing the unfairness of each pattern, remembering the smallest unfairness we have seen so far, and omitting any pattern that has unfairness larger than the smallest seen so far.

Remark 3.1. The above calculation of T_n can be achieved alternatively by a simpler to state but slightly slower algorithm: first generate all $\mathcal{X}$-patterns, and then compute the optimum assignment of each by min-cost flow.

Theorem 3.2. *Algorithm 1 finds the Pareto front of the assignment problem in time $O(kn^{l(k-1)})$, for any metric-based clustering objective and any pattern-based fairness objective.*

Proof Sketch: We prove this theorem by finding an invariant of Algorithm 1: $T_j[P]$ stores the lowest cost assignment of $\mathcal{X}_j$ that maps to pattern P. We maintain this invariant as we build up our table by searching over possible smaller patterns that we can add our next datapoint x_j to and creating the lowest cost assignment that maps to P. Therefore, the assignments stored at $T_n[P]$ are the candidate points for the Pareto front. A simple filtering heuristic, as described above, removes the dominated points from this set of candidates. In terms of running time, the number of possible patterns of total size up to n is upper bounded by $n^{l(k-1)}$, since each of the lk entries of P takes values between 0 and n, and the tuple corresponding to the last cluster k is fully determined by the other clusters. For each considered pattern, we need to look up at most k previous entries of the table T.

3.2 Approximating the Pareto Front for the Clustering Problem

As we saw, Algorithm 1 computes the Pareto front of the Assignment problem exactly for any input centers S. We next show that it also provides an approximation for the Pareto front of the clustering problem. For this purpose, we first use a vanilla clustering algorithm $\mathcal{A}$ for the single-objective

[2] Each clustering $\mathcal{C}$ maps to a pattern $P^{\mathcal{C}}$, with many different clusterings mapping to the same pattern.

problem of minimizing the cost c, to obtain the set S of cluster centers, and then apply Algorithm 1 with this set of centers S. Let α be the approximation ratio of algorithm $\mathcal{A}$.

Definition 3.3 ($\mathcal{W}$-approximation of the Pareto Set for clustering)**.** For parameters $\mathcal{W} = (w_c, w_f)$, we define the $\mathcal{W}$-approximation of the Pareto set X_P as a set of feasible points X'_P such that $\forall x \in X_P, \exists x' \in X'_P$ such that $c(x') \leq w_c \cdot c(x)$ and $f(x') \leq w_f \cdot f(x)$.

This definition is a direct generalization of ϵ-approximate Pareto set defined by Papadimitriou and Yannakakis [55]. One may recover the ϵ-approximate definition by setting $\epsilon = \max(w_c, w_f) - 1$.

Theorem 3.4. *Algorithm 1 finds a $(2 + \alpha, 1)$-approximation of the Pareto set of the clustering problem with a metric-based cost objective c and a pattern-based and mergeable fairness objective f.*

Proof Sketch: We argue that for any clustering map ϕ^* with centers S^* in the Pareto set for clustering, there exists an assignment to the centers S found by an approximate vanilla clustering algorithm that achieves the same or better fairness and at most $(2 + \alpha)$ times the clustering cost. Then, since Algorithm 1 finds the Pareto set of the assignment problem, we are guaranteed the stated approximation. We construct this assignment using a "routing" argument, first introduced in Bera et al. [9]: we create an assignment ϕ' by routing all points in ϕ^* with center $s^* \in S^*$ to the center in S nearest to s^*. Given that the clustering cost is metric-based, we use the triangle inequality on the cost objective to argue that the cost of ϕ' w.r.t. S is not more than $(2 + \alpha)$ times the cost of ϕ^* w.r.t. S^*. Then, we use the mergeability property of the fairness objective to argue that ϕ' has a weakly better fairness than ϕ^*. We can, in fact, modify Algorithm 1 to include non-mergeable fairness objectives, guaranteeing the same approximation ratio and time complexity:

Theorem 3.5. *We can compute a $(2+\alpha, 1)$-approximation of the Pareto set for the clustering problem with a metric-based cost objective c and a pattern-based fairness objective f in time $O(kn^{l(k-1)})$.*

Proof Sketch: The trick here is to transform non-mergeable fairness objectives into mergeable ones, and then apply Algorithm 1. In doing so, we control this transformation through re-assigning the centers of potentially empty clusters (which become an issue in non-mergeable fairness functions). Specifically, for patterns that have empty clusters, we search for the best possible fairness over all ways to reassign the empty clusters' centers to other centers' locations and divide up the points in the non-empty centers. A detailed description and proof for this modified algorithm can be found in Appendix A.

3.3 Intractability

The running time of Algorithm 1 for computing the Pareto front has the parameters k, l in the exponent of n. What evidence do we have that this computation is necessary?

One reason would be that the sheer size of the Pareto front may be exponential. For some objectives, including the BALANCE and the GROUP EGALITARIAN objectives defined in Section 2, the Pareto front is provably of polynomial size. The reason is that all possible values of these objectives are rational numbers involving integers that are all smaller than n, and there are $O(n^2)$ possible different such rational numbers. For other objectives, there may be instances of exponential Pareto fronts, as the objectives that sum over the clusters.

However, a different argument provides a justification of the algorithm's exponential performance: In several papers in recent literature (e.g., see [26, 27]), it is shown that, for a variety of cost functions c and fairness objectives f (including all mentioned proportional violation based objectives), it is NP-hard to find the assignment $\mathcal{C}$ that has the smallest $c(\mathcal{C})$ under the constraint that $f(\mathcal{C}) \leq F$ (for some bound F). We conclude that, unless P = NP, there is no polynomial-time algorithm for outputting the Pareto front, for at least some combinations of c and f. As a footnote to this discussion, the above complexity argument rules out polynomial algorithms, but not exponential algorithms of the form, for example, $O(2^k n^3)$, which is much more benign than $O(n^{(k-1)l})$. These would be ruled out, subject to complexity conjectures, if the problem of finding the least costly clustering subject to fairness constraints were shown to be W-complete, a more severe form of complexity that rules out this more benign exponential performance [23]. We leave this as an interesting open question in the theory of the fairness-cost trade-off of clustering.

3.4 A Polynomial Algorithm for the Pareto Front of the Sum of Imbalances

For *specific* cost and fairness objectives, there is still hope that polynomial-time algorithms exist. As we show below, for a simple objective derived from the BALANCE objective, such an algorithm is possible. For the SUM OF IMBALANCES fairness objective and a clustering objective defined in Section 2, we want to compute the Pareto front when $l = 2$. Surprisingly, it turns out that this problem can be solved in polynomial time by a reduction to the weighted matching problem.

Theorem 3.6. *If $l = 2$ and the fairness objective f is the sum of imbalances $f(\mathcal{C}) = \sum_{i \in [k]} |C_i^1 - C_i^2|$, then the Pareto front of the assignment problem can be computed in polynomial time.*

Proof Sketch: The image of f is contained in the integer set $\{1, \cdots, n\}$. For each potential value j, we construct a graph G_j that contains $\mathcal{X}$ as nodes and another j 'dummy' nodes. We put an edge (u, v) between every $u, v \in \mathcal{X}$ with different sensitive attribute value with weight equal to the cost $\min_i(d^p(u, i) + d^p(v, i))$. Between every data point $u \in \mathcal{X}$ and dummy point v put an edge (u, v) with cost $\min_i d^p(x, s_i)$. Finding the minimum cost perfect matching in this graph gives the minimum cost of an assignment with fairness value j. The same result can be shown similarly for the MAX IMBALANCE objective, $\max_{i \in [k]} |C_i^1 - C_i^2|$ (Theorem A.5).

Remark 3.7. These objectives are quite natural extensions of the BALANCE objective, as they minimize the sum or max, over the k clusters, of the deviation from equality between the two groups. It is worth noting that slight modification of these objectives place the problem back in the NP-complete space: A construction of Bercea et al. [10] (see also Esmaeili et al. [26], Theorem 5.1) implies that for minimizing the deviations not from equality $(1 : 1)$ but from the ratio $1 : 3$, it is NP-complete to find even the point of the Pareto front with the best fairness value.

4 Experiments

We implement our proposed algorithm (Algorithm 1) for finding the Pareto front on three real-world datasets and five fairness objectives, as detailed below.

Datasets: We use the following real-world datasets for our experiments: the Adult dataset and the Census dataset retrieved from the UCI repository (as the Census1990 version) [43, 46], and the BlueBike trip history dataset.[3] The Adult and Census datasets contain numeric attributes such as income, age, education level, often used in applications of fair clustering [6, 19, 26]. The BlueBike dataset contains the starting location, ending location, as well as various user attributes for users of the BlueBike bike sharing system in the Boston area. We use as features the starting and ending longitude and latitude values for all rides during a period of a week in May, 2016. These routes are a proxy for common traffic patterns, for which clustering can inform of high-density areas, with the purpose of deciding on new locations for bike stations or public transportation. Clustering and related framings have long been used in facility location problems, for which fairness is a central question [1, 41, 53, 62]. As the datasets are prohibitively large, we sample from each $1,000$ data points. Further details about the datasets can be found in Appendix C.

Objectives: We implement the k-means clustering objective and five different fairness objectives, as defined in Section 2 (BALANCE, GROUP UTILITARIAN, GROUP UTILITARIAN-SUM, GROUP EGALITARIAN, and GROUP EGALITARIAN-SUM).

Experimental details: We first note that our approach for finding the Pareto front is agnostic to the specific vanilla clustering algorithm used. For our datasets, we use the k-means++ clustering algorithm as the vanilla clustering that has an approximation ratio of $O(log(k))$ [5]. All datasets have numeric attributes, allowing a direct embedding into Euclidean space and using k-means++ directly on the features. We use the self-reported gender (male or female) as the sensitive attribute for all datasets. Each sensitive attribute a has a proportion p_a in the general population. For the proportional violation objectives, we set upper and lower bounds as a δ-deviation from the true proportions $(p_a)_a$: $\alpha_a = (1 + \delta)p_a, \beta_a = (1 - \delta)p_a$. We set $\delta \in [0.005, 0.05]$ for all experiments and $k = 2$ clusters. Additional experiments for $k = 3$ are reported in Appendix E, noting qualitatively similar results. All experiments are run on local computers, using Python 3.9, k-means++ [5], NetworkX [33], and

[3] The BlueBike trip history dataset is retrieved from `https://bluebikes.com/system-data`.

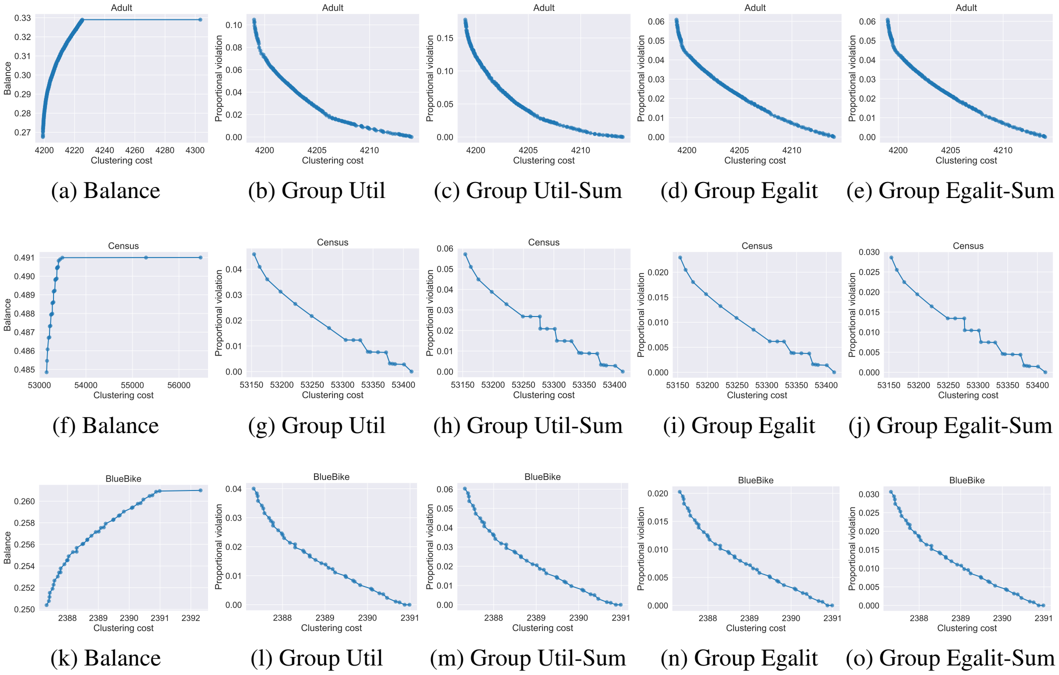
CPLEX for the implementation of the FCBC algorithm [52, 26] with $\epsilon = 2^{-10}$ and $N = 50$ runs. An empirical analysis of the running time for Algorithms 1 and 2 can be found in Section D, Figure 5.[4]

4.1 Pareto Front on Real-World Data

Figure 1 illustrates the Pareto front recovered by our dynamic programming approach (Algorithm 1) on the real-world datasets: the curves obtained are an exact recovery of the Pareto front for the *assignment* problem (as we are not re-computing the clusters centers during the implementation), and thus an approximation for the true Pareto front of the *clustering* problem. We note that the BALANCE objective and the proportional violation objectives differ: higher balance is considered fairer, whereas lower proportional violation is considered fairer, hence the different shapes of the Pareto fronts. From an evaluation point of view, for each assignment found on the Pareto front, we compute its clustering cost with respect to its *actual* centers, rather than the initial centers found by k-means++.

We note that the Pareto front is often, but not always strictly convex or concave, as it simply contains all the undominated points. We note that the proportional violation values will always be worse for the summed objectives than for their min-max equivalents, since the worst proportional violation Δ_a^C is always non-negative, $\forall a \in [l], C \in \mathcal{C}$.

A particular advantage of finding the entire Pareto front is visible for the BALANCE objective in all datasets: as the clustering cost increases, the gain in the BALANCE objective becomes negligible; thus, a practitioner wishing to achieve some level of fairness may gain a lot in quality by allowing BALANCE to decrease by a minimal amount.



(a) Balance (b) Group Util (c) Group Util-Sum (d) Group Egalit (e) Group Egalit-Sum

(f) Balance (g) Group Util (h) Group Util-Sum (i) Group Egalit (j) Group Egalit-Sum

(k) Balance (l) Group Util (m) Group Util-Sum (n) Group Egalit (o) Group Egalit-Sum

Figure 1: Pareto front recovered by Algorithm 1 for the Adult, Census, and BlueBike datasets (by row), for various fairness objectives (by column), for $k = 2$ clusters.

4.2 Exploring Faster Pareto Front Approximations

We explore faster algorithms for recovering the Pareto front for the clustering problem. In particular, we leverage a recently proposed linear programming approach that imposes an upper bound on the clustering objective and optimizes a fairness objective, the FCBC algorithm proposed by Esmaeili et al. [26]. For the GROUP UTILITARIAN and GROUP EGALITARIAN objectives with a clustering

[4] All code and data used in the paper can be found at this link.

cost upper bound input U, FCBC presents a polynomial-time approach for finding an approximation of a point x on the Pareto front that has a clustering cost upper bounded by a quantity $(2+\alpha)U$ with a fairness additive approximation of η. Here, α is the approximation ratio of a vanilla clustering algorithm for the clustering objective, and η is an additive approximation for the fairness objective. Thus, we can extend the $\mathcal{W}$-approximation of a Pareto set definition (Definition 3.3) to include an additive approximation term: for parameters $\mathcal{W} = ((w_c, v_c), (w_f, v_f))$, we can define the $(\mathcal{W}, \mathcal{V})$-approximation of the Pareto set X_P as a set of feasible points X'_P such that $\forall x \in X_P, \exists x' \in X'_P$ such that $c(x') \leq w_c \cdot c(x) + v_c$ and $f(x') \leq w_f \cdot f(x) + v_f$. We note that Algorithm 1 gives only a multiplicative approximation, so $v_c = v_f = 0$. In contrast, for a point $x \in X_P$ on the true Pareto front, FCBC recovers a $((2+\alpha, 0), (1, \eta))$-approximation.[5]

We extend the FCBC algorithm by allowing a sweep over the clustering cost upper bound U, thus, in theory, obtaining an approximation of the Pareto front in polynomial time (for a detailed description, see Algorithm 2 in Appendix D).

Figure 2 shows that repeated FCBC (Algorithm 2) recovers few points on the Pareto front recovered by Algorithm 1: sometimes it recovered the vanilla clustering cost and fairness (the upper most left point in panels b,c,e,f), whereas in other cases it recovers dominated clustering assignments (in panels a-d). We attribute this inaccuracy in recovery to the additive approximation in the fairness objective. Furthermore, whereas both our dynamic programming approach and repeated FCBC have an approximation ratio of $(2+\alpha)$ in the clustering objective, dynamic programming often gets a strictly better cost for similar fairness values. In other words, where repeated FCBC gains in running time,[6] it loses in recovery accuracy and clustering cost. This is particularly problematic when the only point recovered by FCBC is the vanilla clustering assignment: this means that even when a practitioner may be willing to trade-off significant clustering cost in order to improve fairness, that trade-off is not realizable in practice solely through FCBC.

Furthermore, the FCBC algorithm does not work for objectives summing over the clusters, such as GROUP UTILITARIAN-SUM and GROUP EGALITARIAN-SUM. For such objectives, there is no polynomial time algorithm that can recover the Pareto front to within an additive approximation of $O(n^{\delta})$, for $\delta \in [0, 1)$ (see Theorem 7.1 in Esmaeili et al. [26]). Our approach, however, can also include such objectives, as we show both in theory and in practice.

5 Discussion

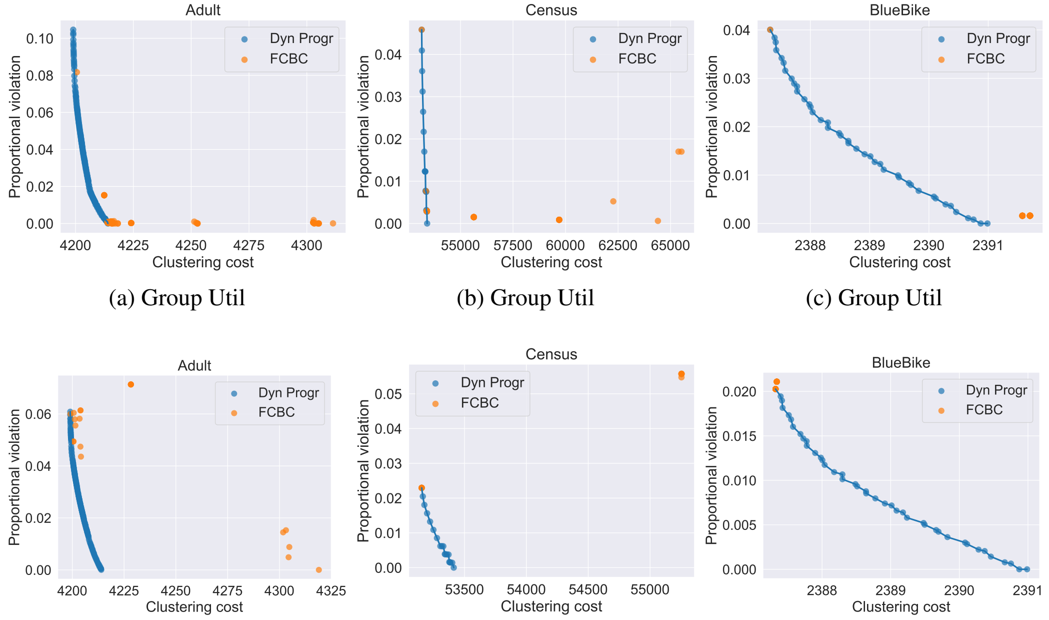
Overall, our work shows the versatility of our proposed algorithms for a variety of fairness and clustering objectives. We provide simple properties sufficient in theory for the recovery of the Pareto front for the (clustering, fairness)-biobjective optimization problem, with extensive experiments on real-world datasets. We discuss limitations and future directions of our work in this section.

First, while our approach has the advantage of being agnostic to specific objectives, it loses in running time as compared to approaches that optimize for specific fairness objectives. While previous work provides approximation algorithms for optimization clustering under a fairness constraint, none offers a method for computing the Pareto front. On the question 'can we have faster algorithms with worse approximation bounds?', the answer is yes. Our paper provides (the first, to our knowledge) such exploration: We show in Section 4.2 that we can adapt such polynomial-time methods to compute an approximation of the Pareto front; however, what we gain in runtime, we lose in approximation bounds: the repeated FCBC algorithm has an additive approximation on the fairness objective that can get large in practice (we get a different Pareto front, with points quite far away in the objective cost, compared to the dynamic programming approach), and with no theoretical bounds on the additive approximation.

A future direction could study faster algorithms that can recover a similar approximation of the Pareto front. Interesting theoretical open questions emerge: while our analysis ruled out polynomial algorithms for general objectives, are there exponential algorithms (i.e. of the form $O(2^k n^3)$), that still provide an improvement from $O(n^{(k-1)l})$-type of algorithms? And, are there other objectives

[5] In particular, η is dependent on the fairness objective, on U, and on the specific instance of the fair clustering problem (see Theorem 6.1 in Esmaeili et al. [26]). A potential limitation of this approximation is that η is not efficiently computable in closed-form for a particular instance.

[6] The FCBC algorithm uses linear programming through the simplex method, with a worst-case running time of $O(2^n)$; in practice, it is faster than that, as noted in the Appendix.



(a) Group Util (b) Group Util (c) Group Util

(d) Group Egalit (e) Group Egalit (f) Group Egalit

Figure 2: Pareto front recovered by Algorithm 1 (labeled ‘Dyn Progr’, blue) and by Algorithm 2 (labeled ‘FCBC’, orange), for $k = 2$ clusters.

that admit polynomial-time algorithms for computing an approximate Pareto, aside from SUM/MAX OF IMBALANCES? Furthermore, future work could investigate algorithms for recovering the Pareto front for fairness objectives that are not pattern-based, which is a prerequisite for our algorithms. We provide some examples of such objectives in the Appendix.

Our approach is best suited for group fairness definitions. Extending this work to other definitions would be an excellent avenue for future work. We note that for some individual fairness definitions, the notion of a Pareto front does not apply: for example, the socially fair k-means clustering notion proposed by Ghadiri et al. [31] defines a single optimization objective that already incorporates fairness (by minimizing the max distance between points in each group and their cluster centers). With this definition, the fairness objective and the clustering objective are not separated in a multi-objective optimization problem, but rather combined in a single objective. Therefore, there is no Pareto front, since this notion applies to a multi-objective optimization problem. For other definitions of individual fairness, such as recent adaptations of individual fairness under bi-criteria optimization problems [50], one can apply a repeated approximation algorithm under an upper bound on one of the objectives (similar to the repeated-FCBC algorithm). In doing so, one would obtain loose approximation guarantees (see, for example, the approximation guarantee for individual fairness proved by Mahabadi and Vakilian [50]). Dynamic programming approaches would not directly apply; however, we think this would be an excellent avenue for future work.

Finally, we assumed that the sensitive attributes are disjoint. For overlapping attributes, one could assign a new sensitive attribute to every overlap set and apply our approach, however, with a significant increase in running time. Future work could investigate alternative approaches that optimize running time for overlapping attributes.

Acknowledgments and Disclosure of Funding

The authors would like to thank Vivian Y. Nastl for valuable feedback on earlier drafts of the paper. A.-A. S. acknowledges support from the Tübingen AI Center. M. Y. acknowledges support from NSF grants CCF-2332922 and CCF-2212233.

References

[1] Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. Fair clustering via equitable group representations. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 504–514, 2021.

[2] Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. Clustering without over-representation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 267–275, 2019.

[3] Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. *SIAM Journal on Computing*, 49 (4):FOCS17–97, 2019.

[4] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machine learning*, 75:245–248, 2009.

[5] David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms*, volume 7, pages 1027–1035, 2007.

[6] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. Scalable fair clustering. In *International Conference on Machine Learning*, pages 405–413. PMLR, 2019.

[7] MohammadHossein Bateni, Vincent Cohen-Addad, Alessandro Epasto, and Silvio Lattanzi. A scalable algorithm for individually fair k-means clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 3151–3159. PMLR, 2024.

[8] Cristina Bazgan, Laurent Gourvès, and Jérôme Monnot. Approximation with a fixed number of solutions of some multiobjective maximization problems. *Journal of Discrete Algorithms*, 22: 19–29, 2013.

[9] Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. *Advances in Neural Information Processing Systems*, 32, 2019.

[10] Ioana O Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R Schmidt, and Melanie Schmidt. On the cost of essentially fair clusterings. *arXiv preprint arXiv:1811.10319*, 2018.

[11] Fritz Bökler and Markus Chimani. Approximating multiobjective shortest path in practice. In *2020 Proceedings of the Twenty-Second Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 120–133. SIAM, 2020.

[12] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[13] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of computational social choice*. Cambridge University Press, 2016.

[14] Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Transactions on Algorithms (TALG)*, 13(2):1–31, 2017.

[15] Darshan Chakrabarti, John P Dickerson, Seyed A Esmaeili, Aravind Srinivasan, and Leonidas Tsepenekas. A new notion of individually fair clustering: α-equitable k-center. In *International Conference on Artificial Intelligence and Statistics*, pages 6387–6408. PMLR, 2022.

[16] Weiyu Chen and James Kwok. Multi-objective deep learning with adaptive reference vectors. *Advances in Neural Information Processing Systems*, 35:32723–32735, 2022.

[17] Xingyu Chen, Brandon Fain, Liang Lyu, and Kamesh Munagala. Proportionally fair clustering. In *International Conference on Machine Learning*, pages 1032–1041. PMLR, 2019.

[18] Anshuman Chhabra, Karina Masalkovaitė, and Prasant Mohapatra. An overview of fairness in clustering. *IEEE Access*, 9:130698–130720, 2021.

[19] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. *Advances in Neural Information Processing Systems*, 30, 2017.

[20] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[21] Ilias Diakonikolas and Mihalis Yannakakis. Small approximate Pareto sets for biobjective shortest paths and other problems. *SIAM Journal on Computing*, 39(4):1340–1371, 2010.

[22] John Dickerson, Seyed Esmaeili, Jamie H Morgenstern, and Claire Jie Zhang. Doubly constrained fair clustering. *Advances in Neural Information Processing Systems*, 36, 2024.

[23] Rodney G Downey and Michael Ralph Fellows. *Parameterized complexity*. Springer Science & Business Media, 2012.

[24] Jack Edmonds. Maximum matching and a polyhedron with 0,1 vertices. *J. Res. Nat. Bur. Standards*, 69B:125–130, 1965.

[25] Thomas Erlebach, Hans Kellerer, and Ulrich Pferschy. Approximating multiobjective knapsack problems. *Management Science*, 48(12):1603–1612, 2002.

[26] Seyed Esmaeili, Brian Brubach, Aravind Srinivasan, and John Dickerson. Fair clustering under a bounded cost. *Advances in Neural Information Processing Systems*, 34:14345–14357, 2021.

[27] Seyed A Esmaeili, Sharmila Duppala, John P Dickerson, and Brian Brubach. Fair labeled clustering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 327–335, 2022.

[28] Reza Zanjirani Farahani, Maryam SteadieSeifi, and Nasrin Asgari. Multiple criteria facility location problems: A survey. *Applied mathematical modelling*, 34(7):1689–1709, 2010.

[29] Laurie E Felland, Johanna R Lauer, and Peter J Cunningham. *Suburban poverty and the health care safety net*. Center for Studying Health System Change Washington (DC), 2009.

[30] Harold N. Gabow. Data structures for weighted matching and extensions to b-matching and f-factors. *ACM Trans. Algorithms*, 14(3):39:1–39:80, 2018.

[31] Mehrdad Ghadiri, Samira Samadi, and Santosh Vempala. Socially fair k-means clustering. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 438–448, 2021.

[32] Shivam Gupta, Ganesh Ghalme, Narayanan C Krishnan, and Shweta Jain. Efficient algorithms for fair clustering with a new notion of fairness. *Data Mining and Knowledge Discovery*, 37(5): 1959–1997, 2023.

[33] Aric Hagberg, Dan Schult, Pieter Swart, D Conway, L Séguin-Charbonneau, C Ellison, B Edwards, and J Torrents. Networkx. high productivity software for complex networks. *Webová strá nka https://networkx. lanl. gov/wiki*, 2013.

[34] Vahid Hajipour, Parviz Fattahi, Madjid Tavana, and Debora Di Caprio. Multi-objective multi-layer congested facility location-allocation problem optimization with Pareto-based meta-heuristics. *Applied Mathematical Modelling*, 40(7-8), 2016.

[35] SI Harewood. Emergency ambulance deployment in barbados: a multi-objective approach. *Journal of the Operational Research Society*, 53(2):185–192, 2002.

[36] Mordechai I Henig. The shortest path problem with two objective functions. *European Journal of Operational Research*, 25(2):281–291, 1986.

[37] Long P Hoang, Dung D Le, Tran Anh Tuan, and Tran Ngoc Thang. Improving Pareto front learning via multi-sample hypernetworks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7875–7883, 2023.

[38] Dorit S Hochbaum and David B Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM (JACM)*, 33(3):533–550, 1986.

[39] Bo Huang, P Fery, L Xue, and Y Wang. Seeking the pareto front for multiobjective spatial optimization problems. *International Journal of Geographical Information Science*, 22(5): 507–526, 2008.

[40] Lingxiao Huang, Shaofeng Jiang, and Nisheeth Vishnoi. Coresets for clustering with fairness constraints. *Advances in Neural Information Processing Systems*, 32, 2019.

[41] Christopher Jung, Sampath Kannan, and Neil Lutz. A center in your neighborhood: Fairness in facility location. *arXiv preprint arXiv:1908.09041*, 2019.

[42] Kathrin Klamroth and Margaret M Wiecek. Dynamic programming approaches to the multiple criteria knapsack problem. *Naval Research Logistics (NRL)*, 47(1):57–76, 2000.

[43] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, volume 96, pages 202–207, 1996.

[44] Michael M Kostreva and Malgorzata M Wiecek. Time dependency in multiple objective dynamic programming. *Journal of Mathematical Analysis and Applications*, 173:289–289, 1993.

[45] Bo Li, Lijun Li, Ankang Sun, Chenhao Wang, and Yingfan Wang. Approximate group fairness for clustering. In *International Conference on Machine Learning*, pages 6381–6391. PMLR, 2021.

[46] Moshe Lichman. UCI machine learning repository, 2013.

[47] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[48] Xi Lin, Zhiyuan Yang, Qingfu Zhang, and Sam Kwong. Controllable Pareto multi-task learning. *arXiv preprint arXiv:2010.06313*, 2020.

[49] Suyun Liu and Luis Nunes Vicente. The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. *Annals of Operations Research*, pages 1–30, 2021.

[50] Sepideh Mahabadi and Ali Vakilian. Individual fairness for k-clustering. In *International Conference on Machine Learning*, pages 6586–6596. PMLR, 2020.

[51] Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in Pareto optimization. In *International Conference on Machine Learning*, pages 6597–6607. PMLR, 2020.

[52] IBM CPLEX User's Manual. Version 12 release 7. *IBM ILOG CPLEX Optimization*, 2016.

[53] Michael T Marsh and David A Schilling. Equity measurement in facility location analysis: A review and framework. *European Journal of Operational Research*, 74(1):1–17, 1994.

[54] Maryam Negahbani and Deeparnab Chakrabarty. Better algorithms for individually fair k-clustering. *Advances in Neural Information Processing Systems*, 34:13340–13351, 2021.

[55] Christos H Papadimitriou and Mihalis Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 86–92. IEEE, 2000.

[56] Konstantinos E Parsopoulos and Michael N Vrahatis. Particle swarm optimization method in multiobjective problems. In *Proceedings of the 2002 ACM Symposium on Applied Computing*, pages 603–607, 2002.

[57] John Preston and Fiona Rajé. Accessibility, mobility and transport-related social exclusion. *Journal of transport geography*, 15(3):151–160, 2007.

[58] Seyed Habib A Rahmati, Vahid Hajipour, and Seyed Taghi Akhavan Niaki. A soft-computing Pareto-based meta-heuristic algorithm for a multi-objective multi-server facility location problem. *Applied Soft Computing*, 13(4):1728–1740, 2013.

[59] Juana L Redondo, José Fernández, José Domingo Álvarez Hervás, Aránzazu Gila Arrondo, and Pilar M Ortigosa. Approximating the Pareto-front of a planar bi-objective competitive facility location and design problem. *Computers & Operations Research*, 62:337–349, 2015.

[60] Michael Ruchte and Josif Grabocka. Scalable Pareto front approximation for deep multi-objective learning. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1306–1311. IEEE, 2021.

[61] Hisashi Tamaki, Hajime Kita, and Shigenobu Kobayashi. Multi-objective optimization by genetic algorithms: A review. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 517–522. IEEE, 1996.

[62] Ali Vakilian and Mustafa Yalciner. Improved approximation algorithms for individually fair clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 8758–8779. PMLR, 2022.

[63] Bernardo Villarreal and Mark H Karwan. Multicriteria integer programming: A (hybrid) dynamic programming recursive approach. *Mathematical programming*, 21:204–223, 1981.

[64] Arthur Warburton. Approximation of Pareto optima in multiple-objective, shortest-path problems. *Operations research*, 35(1):70–79, 1987.

[65] Qingfu Zhang and Hui Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

A Complete Proofs

Proof of Theorem 3.2: Define V_P as the set of clusterings $\mathcal{C}$ that have the pettern $P^{\mathcal{C}} = P$ and $\mathcal{C}$ is a clustering of $\mathcal{X}_j$. Let c be the metric-based cost function for the assignment to centers S, which is parameterized by a distance function d and non-negative exponent p. We will argue that, for all j, the clustering $\mathcal{C}$ stored at $T_j[P]$ has the property that $\mathcal{C} = \arg\min_{\mathcal{C}' \in V_P} c(\mathcal{C}')$. We will show that this invariant always holds by induction on the number of data points in the current level of the table j (equivalently, the number of points clustered by $\mathcal{C}$ in the main loop of Algorithm 1). The base case is $j = 0$, where we store the empty clustering, the unique clustering of 0 points.

Suppose we have some pattern P for the set of the first j points in our ordering, $\mathcal{X}_j$. Let $\mathcal{C}^* = \arg\min_{\mathcal{C} \in V_P} c(\mathcal{C})$, so $\mathcal{C}^*$ is optimal cost for P. Point x_j must be in some cluster $C_i \in \mathcal{C}^*$. The clustering $\mathcal{C}'_i$ of the remaining points (which are exactly the first $j-1$ points from the data) induces some pattern $P'_i = P^{\mathcal{C}'_i}$ which is an $\mathcal{X}_{j-1}$-pattern. Since $\mathcal{C}^*$ was assumed to be of optimal quality, the clustering $\mathcal{C}'_i$ must have the property that $\mathcal{C}'_i = \arg\min_{\mathcal{C} \in V_{P'_i}} c(\mathcal{C})$. Otherwise, $\mathcal{C}'_i$ could be replaced in $\mathcal{C}^*$ with the lower cost clustering that has the same pattern and reduce the cost of $\mathcal{C}^*$ (a contradiction).

We have shown that $\mathcal{C}^* = \mathcal{C}_i$, where $\mathcal{C}_i = \mathcal{C}'_i + x_j$ is generated by assigning the first $j-1$ points according to $\mathcal{C}'_i$, and then adding point x_j to cluster i. By the inductive assumption $T_{j-1}[P'_i]$ correctly stores the optimal $\mathcal{C}'_i$.

Since Algorithm 1 minimizes $c(\mathcal{C}_i)$ over cluster indices i, we correctly compute the optimal clustering of j points that maps to P. All candidate points for the Pareto set of the assignment problem are stored in T_n, meaning that every clustering $\mathcal{C}'$ of $\mathcal{X}$ is weakly dominated by some clustering in T_n. This is because every clustering $\mathcal{C} \in \mathcal{K}$ must map to some $\mathcal{X}$-pattern P. By our algorithm invariant, $\mathcal{C}^* \in T_n[P]$ is the clustering that minimizes assignment cost across clusterings that have pattern P, and thus $c(\mathcal{C}^*) \leq c(\mathcal{C})$. In addition, since $P^{\mathcal{C}^*} = P^{\mathcal{C}'}$, and f is a pattern-based fairness objective, $f(\mathcal{C}^*) = f(\mathcal{C})$. So, $\mathcal{C}$ is weakly dominated by $\mathcal{C}^* \in T_n$. Therefore, filtering dominated points out of the clusterings in T_n gives us the complete Pareto set for the assignment problem with pattern-based fairness objectives and metric-based cost objectives on n data points. Note that the output of Algorithm 1 recovers the Pareto set $\{\mathcal{C}\}$ as well as the Pareto front $\{(c(\mathcal{C}), f(\mathcal{C})\}$, since the Pareto front is just the image of the Pareto set under the two objectives. □

Proof of Theorem 3.4: Theorem 3.2 states that Algorithm 1 finds (exactly) the Pareto front of the assignment problem. Let ϕ^*, S^* be the optimal cost clustering map and set of centers, respectively, that satisfy a particular fairness bound F. Let ϕ, S be the clustering map and set of centers found by a vanilla clustering algorithm $\mathcal{A}$, which achieves an α-approximation to the best clustering cost. We define a new assignment ϕ', S, by applying a "routing" argument, first introduced in Bera et al. [9] and reused in Esmaeili et al. [27].

Define a function $\text{nrst}(s^*) = \arg\min_{s \in S} d^p(s, s^*)$ which returns the nearest center in S to an input center s^*. Now define an assignment map ϕ' where vertices are routed from their center $s_i^* \in S^*$ to $\text{nrst}(s_i^*) \in S$. In other words, for every point x, $\phi'(x) = \text{nrst}(\phi^*(x))$. Now we can write:

$$\begin{aligned} d^p(x, \phi'(x)) = d^p(x, \text{nrst}(\phi^*(x))) &\leq d^p(x, \phi^*(x)) + d^p(\phi^*(x), \text{nrst}(\phi^*(x))) \\ &\leq d^p(x, \phi^*(x)) + d^p(\phi^*(x), \phi(x)) \\ &\leq 2d^p(x, \phi^*(x)) + d^p(x, \phi(x)) \end{aligned}$$

The first and third inequalities follow from the triangle inequality on d^p and the second inequality is due to the definition of the nrst function. In addition, $\alpha \left(\sum_x d^p(x, \phi^*(x))\right)^{1/p} \geq \left(\sum_x d^p(x, \phi(x))\right)^{1/p}$, since ϕ is the clustering found by the vanilla clustering algorithm $\mathcal{A}$. Applying our previous inequality together with the triangle inequality on the p-norm:

$$\begin{aligned} \left(\sum_x d^p(x, \phi'(x)\right)^{1/p} &\leq \left(\sum_x 2d^p(x, \phi^*(x))\right)^{1/p} + \left(\sum_x d^p(x, \phi(x))\right)^{1/p} \\ &\leq (2 + \alpha) \left(\sum_x d^p(x, \phi^*(x))\right)^{1/p} \end{aligned}$$

This implies that $\phi' S$ is an assignment map with respect to centers S that has at most $(2+\alpha)$ times the cost of ϕ^* with respect to centers S^*.

In addition, the clustering $\mathcal{C}'$ associated with ϕ' can be generated by, for each $s_j \in S$, merging all clusters $C_i^* \in \mathcal{C}^*$ such that $\text{nrst}(s_i^*) = s_j$. This procedure can be done by sequentially merging pairs of clusters. Since the fairness objective is mergeable (see Def. 2.2), this implies that $f(\mathcal{C}^*) \geq f(\mathcal{C}')$. So, the assignment map ϕ' with respect to centers S also satisfies the fairness bound F. There exists an assignment to centers S that is a $(2+\alpha, 1)$-approximation to ϕ^*, S^*, so the Pareto front of the assignment problem is a $(2+\alpha, 1)$-approximation to the Pareto front of the clustering problem as desired. □

Proof of Theorem 3.5. We describe in detail the algorithm modification needed to compute the Pareto front for non-mergeable fairness objectives. First, we give some necessary preliminaries:

Definition A.1 (Refinement of a pattern and refinement DAG D)**.** Consider the directed graph D with the $\mathcal{X}$-patterns as nodes and edges (P_1, P_2) if merging two nonempty rows of the pattern P_2 yields the pattern P_1. A pattern P' is a refinement of another pattern P iff there is a path from P to P' in D, i.e. if P can be obtained from P' by merging different parts of P'. Note that D is a directed acyclic graph (DAG). Let R_P be the set of P' that are refinements of P.

Similarly, one can define the refinement of a clustering $\mathcal{C}$. Note that if a clustering $\mathcal{C}'$ is a refinement of another clustering $\mathcal{C}$, then its pattern $P^{\mathcal{C}'}$ is a refinement of the pattern of cluster $\mathcal{C}$, $P^{\mathcal{C}}$.

Next, we define a modified fairness function created from the non-mergeable function f, with the purpose of reducing it to a mergeable function and applying Algorithm 1.

Definition A.2 (Modified fairness function)**.** If f is a pattern-based fairness function, define its associated modified function $\hat{f}$ to be $\hat{f}(P) = \min_{P' \in R_P}(f(P'))$ for every pattern P.

Note that f is still required to be pattern-based, and thus modified function is well-defined. By definition, $\hat{f}$ is a mergeable function. Furthermore, if f is mergeable then $\hat{f} = f$.

Lemma A.3. *We can compute in $O(n^{l(k-1)})$ time the modified function $\hat{f}$ for all $\mathcal{X}$-patterns P, and compute for each P a refinement P' such that $\hat{f}(P) = f(P')$.*

Proof. We compute $\hat{f}$ bottom up in the DAG D. We also compute for each node a pointer to its refinement pattern that has the minimum fairness cost. At the sinks P (patterns that have no outward edge pointing from them to other patterns), $\hat{f}(P) = f(P)$ and P points to itself only. For every non-sink node P, we set $\hat{f}(P) = \min_{v \in N(P)} \hat{f}(v)$ where $N(P)$ is the set of neighbors of P, and we set the pointer of P to the descendant pointed by the neighbor v of P with the minimum $\hat{f}(v)$.

The total time complexity is linear in the number of nodes and edges of D. The number of edges is at most k^2 times the number of nodes, since every node P' has at most k^2 incoming edges (there are at most k^2 choices for the two parts of P' that are merged to form a parent pattern). The number of nodes of D, i.e. $\mathcal{X}$-patterns, is at most $4k(\frac{n}{2})^{l(k-1)}$: All the components of all the rows of a pattern P, except possibly at most two components, are less than $n/2$; furthermore, specifying $k-1$ rows of P determines also the last row because the sum for each attribute value must match the given set of points. From the bounds on the number of nodes and edges of D, it follows that the time complexity is $O(n^{l(k-1)})$. □

Recall that in a clustering we associate a center with each cluster, and the cost of a clustering is computed from the distances of the points from the center of their cluster. We allow different clusters to have the same point as their center. We show the following result, needed in the proof of Theorem 3.5:

Lemma A.4. *If we are given a clustering $\mathcal{C}$ with centers S and P' is a refinement of $P^{\mathcal{C}}$ then we can compute efficiently a clustering $\mathcal{C}'$ with centers S', which we call a center reassignment of $\mathcal{C}$, such that $P^{\mathcal{C}'} = P'$ and $c(\mathcal{C}', S') = c(\mathcal{C}, S)$. The centers S' will be a subset of the points in S but with multiplicity (i.e. different clusters may have the same center).*

Proof. Suppose that row i of pattern P is formed by merging a set J_i of two or more nonempty parts of P'. Then we split the cluster C_i of $\mathcal{C}$ into a set of $|J_i|$ subclusters, one for each part in J_i, and we place in each subcluster a number of points from C_i for each attribute value that matches the corresponding entry in the row of P'; we assign to all the subclusters the same center as the center of C_i.

After performing the above splitting for all parts i of P that are refined in the pattern P', we obtain a clustering $\mathcal{C}'$ such that $P^{\mathcal{C}'} = P'$. Since the subclusters created from splitting a cluster C_i of $\mathcal{C}$ are assigned the same center as the center of C_i, it follows from the definition of a metric-based cost function, that $\mathcal{C}$ and $\mathcal{C}'$ have the same cost.

□

We now describe the algorithm modification for non-mergeable fairness objectives. For a non-mergeable fairness objective f, let $\hat{f} = \min_{P' \in R_P}(f(P'))$ be its associated modified function. Apply the algorithm of Lemma A.3 to compute $\hat{f}(P)$ for every $\mathcal{X}$-pattern P, and the corresponding pointer to its optimal refinement P'. As before, use a vanilla clustering approximation algorithm $\mathcal{A}$ to compute a clustering that approximates the minimum cost. Use the centers of this clustering in Algorithm 1 to construct the dynamic programming table T_n. Process the patterns P as before in order of increasing cost, but now use the modified function $\hat{f}$ as the fairness objective to filter out dominated patterns, and for each undominated pattern P, replace the clustering $\mathcal{C}$ in $T_n[P]$ by the center reassignment clustering $\mathcal{C}'$ constructed as in Lemma A.4. The algorithm returns the set of these center reassignment clusterings $\mathcal{C}'$ for all undominated patterns. By the definition of the modified fairness function and Lemma A.4, these are undominated clusterings for the original (non-mergeable) fairness function f and the clustering cost c.

From our timing analysis, the time complexity of the algorithm is $O(kn^{l(k-1)})$.

The proof of the approximation follows the proof of Theorem 3.4. We define ϕ^*, S^*, ϕ, S, and ϕ', S similarly as before. Observe that in the construction of ϕ', we simply merged clusters from ϕ^* and gave them new centers. Since the pattern of a clustering is independent of the identities of the centers, the pattern of ϕ^* is a refinement of the pattern of ϕ'. So by Lemma A.4, there exists a clustering ϕ'' that is a center reassignment of ϕ' and has the same pattern as ϕ^*.

Therefore, ϕ'', which is one of the reassignments we search over in the algorithm, has $f(\mathcal{C}'') = f(\phi^*)$ and $c(\mathcal{C}'') = c(\phi')$ (here, we used interchangeably the clustering cost function c applied to the clustering or the clustering assignment map, which are equivalent). Since ϕ' has at most $2 + \alpha$ worse cost than ϕ^* (see proof of Theorem 3.4), so does $\mathcal{C}''$. Therefore, the Pareto front computed by the algorithm is a $(2 + \alpha, 1)$-approximation to the Pareto front of the clustering problem, as desired. □

Proof of Theorem 3.6: First we notice that in this case f can take at most $n + 1$ values: If $n = |\mathcal{X}|$ is even, then it can take only values the even integers between 0 and n, and if it is odd all odd integers between 1 and $n - 1$. We shall treat the case of even n, the case of even n being very similar.

Given dataset $\mathcal{X}$, metric d, and exponent p, for each even number F between 0 and n we must compute the best assignment that has fairness F. Once this is done, we only have to sort with respect to fairness and remove the dominated assignments (in a similar vein to the filtering heuristic of Algorithm 1). Clearly, this step is polynomial time since we can sort the points in $O(n \log(n))$ and then make a single pass over them to remove the dominated points. All that remains to show is that we can compute the best assignment with fairness F also in polynomial time. We do so as follows.

Given an even number F, we construct a weighted graph G_F with n nodes corresponding to the data points, plus F *dummy nodes.* We join every data point x with every dummy node z by an edge of weight $\min_{i \in [k]} d^p(x, s_i)$. We join any two data points x, y with different attribute value (recall that there are only two attribute values) by an edge of length $\min_{i=1}^{k}(d^p(x, s_i) + d^p(y, s_i))$. We call an edge of G_F to be *of type* $i \in [k]$ if the i-th cluster achieves the minimum that defines the weight of the edge. (In other words, i is the argmin of the minimization expression.)

A perfect matching in a graph is a set of disjoint edges that includes all the nodes. The weight of a matching M is defined as the sum of the weights of edges that are contained in the matching: $w(M) = \sum_{e \in M} w(e)$. Compute the minimum weight perfect matching $\hat{M}$ in the graph G_F. This can be done in polynomial time using Edmonds' algorithm [24], specifically in time $O(N(E+N\log N))$ for a graph with N nodes and E edges [30]; in our case the graph G_F has at most $2n$ nodes and $2n^2$ edges.

We will show that the minimum weight of a perfect matching is equal to the minimum cost of a clustering with fairness F, and furthermore we can derive a minimum cost clustering from the minimum weight matching $\hat{M}$.

Consider any clustering $\mathcal{C}$ with $f(\mathcal{C}) = F$. Starting from $\mathcal{C}$, we construct a perfect matching M of G_F as follows: For each cluster $C \in \mathcal{C}$, choose the attribute value a that has fewer nodes in C (breaking ties arbitrarily). Now, match these nodes of attribute a in C arbitrarily with nodes of the larger attribute group in C. Finally, match any remaining nodes of the other group (the larger attribute group) to dummy nodes.

We claim that such a matching M is possible, because the total number of nodes that cannot be matched is precisely F, the number of dummy nodes. We claim now that the weight of this matching $w(M)$ is at most the cost of the clustering, $c(\mathcal{C})$; that is, $w(M) \leq c(\mathcal{C})$.

The weight of the matching is at most the clustering cost for the following reason: each matched pair x, y contributes to $c(\mathcal{C})$ at least the weight of the corresponding matched edge; and any data point matched to a dummy node again contributes to $c(\mathcal{C})$ at least the weight of the matched edge.

Now consider the minimum weight perfect matching $\hat{M}$ of the weighted graph G_F. Clearly, $w(\hat{M}) \leq c(\mathcal{C})$. Construct from $\hat{M}$ a corresponding clustering $\hat{\mathcal{C}}$ as follows: any matched data point whose edge is of type i is placed in cluster i, while any data point matched in $\hat{M}$ with a dummy node with an edge of type i is added to cluster i. This clustering satisfies $c(\hat{\mathcal{C}}) = w(\hat{M}) \leq c(\mathcal{C})$. Since $\mathcal{C}$ was assumed to be an arbitrary clustering with $f(\mathcal{C}) = F$, $\hat{\mathcal{C}}$ is the optimum such clustering. □

The same result can be shown for the MAX IMBALANCE objective.

Theorem A.5. *If $l = 2$ and the fairness objective f is the max imbalance $f(\mathcal{C}) = \max_{i\in[k]} ||C_i^1| - |C_i^2||$, then the Pareto front of the assignment problem can be computed in polynomial time.*

Proof. The objective can take again at most $n+1$ values (more precisely, at most $1+\max(|\mathcal{X}^1|, |\mathcal{X}^2|)$ values). For each possible value F, construct a weighted graph G_F, whose set N_F of nodes consists of n nodes corresponding to the given set $\mathcal{X}$ of data points, a set D_i of F dummy nodes for each $i \in [k]$, and if $n + kF$ is odd, then N_F has one more dummy node so that the total number of nodes is even. The edge set E_F of G_F consists of the following edges: for each pair x, y of data points with different attribute values, we include an edge (x, y) with weight $\min_{i=1}^{k}(d^p(x, s_i) + d^p(y, s_i))$ and associate with the edge as its type the index $i \in [k]$ that achieves the minimum in the weight; for each data point x and dummy node z in D_i, we include an edge (x, z) with weight $d^p(x, s_i)$ and associate type i to the edge; for every pair z, w of dummy nodes we include an edge (z, w) with weight 0 (we do not associate a type with these edges).

We then compute the minimum weight perfect matching $\hat{M}$ of the graph G_F. Just like in the proof of Theorem 3.6, the minimum weight perfect matching can be found in polynomial time. The matching $\hat{M}$ induces an assignment $\hat{\mathcal{C}}$ of the data points: every point x is assigned to the cluster i corresponding to the type of the edge of $\hat{M}$ incident to x. By construction, the cost of the assignment is equal to the weight $w(\hat{M})$ of the matching. Furthermore, for every cluster $i \in [k]$, the number of data points in the cluster that are matched with dummy nodes in D_i is at most F, hence $||\hat{C}_i^1| - |\hat{C}_i^2|| \leq F$. Therefore the fairness cost of the assignment is at most F.

Conversely, any assignment $\mathcal{C}$ with fairness F induces a perfect matching M of G_F, as follows: For each cluster i, match data points of cluster C_i in $\mathcal{C}$ with opposite attribute values arbitrarily in pairs, and match the remaining $||C_i^1| - |C_i^2||$ data points to dummy nodes in D_i. The rest of the dummy nodes that are not matched to a data point are matched arbitrarily in pairs. The weight of this matching M is then at most the cost of the clustering $\mathcal{C}$, $w(M) \leq c(\mathcal{C})$. Therefore, $c(\mathcal{C}) \geq w(\hat{M}) = c(\hat{\mathcal{C}})$, that is, any assignment $\mathcal{C}$ with fairness F is dominated by the assignment $\hat{\mathcal{C}}$ that we derived from the minimum weight perfect matching. □

B Analyzing Fairness Objectives

B.1 Balance and proportional violation-based objectives are pattern-based and mergeable

Proposition B.1. *The fairness objectives* Balance, Sum of Imbalances, Group Utilitarian, Group Utilitarian-Sum, Group Egalitarian, *and* Group Egalitarian-Sum *are pattern-based and mergeable.*

Proof. Balance-based Objectives: As defined in equation 1, the Balance objective is always pattern-based by definition. As above, say that we have two sensitive attributes (as this objective is originally defined for $l = 2$), called R and B. Then, for a cluster $C \in \mathcal{C}$, $|C^R|$ and $|C^B|$ are the number of R and B data points in cluster C, respectively. For two clusterings $\mathcal{C}$ and $\mathcal{C}'$ that induce the same pattern p, it means that we can pair each $C \in \mathcal{C}$ with a different cluster $C' \in \mathcal{C}'$ such that $|C^R| = |C'^R|$ and $|C^B| = |C'^B|$. Thus $\textsc{Balance}(C) = \textsc{Balance}(C')$ for all such pairs, and thus $\textsc{Balance}(\mathcal{C}) = \textsc{Balance}(\mathcal{C})'$.

We show that Balance is also mergeable through a simple induction over the number of clusters to be merged. For the base case, take a clustering $\mathcal{C}$ and construct a clustering $\mathcal{C}'$ in which two clusters from $\mathcal{C}$ have been merged (assume without loss of generality that $\mathcal{C} = \{C_1, C_2, \cdots, C_k\}$ and $\mathcal{C}' = \{\emptyset, C_1 \cup C_2, \cdots, C_k\}$). For ease of notation, denote by $a = |C_1^R|, b = |C_1^B|, c = |C_2^R|, d = |C_2^B|$. Then, $a + c = |(C_1 \cup C_2)^R|, b + d = |(C_1 \cup C_2)^B|$. Note that all variables are non-negative. We assume that neither C_1 and C_2 are empty. We need to show that the fairness objective is at least as good for the merged clustering than for the original clustering. It is sufficient to show that

$$\min\left(\min\left(\frac{a}{b}, \frac{b}{a}\right), \min\left(\frac{c}{d}, \frac{d}{c}\right)\right) \leq \min\left(\frac{a+c}{b+d}, \frac{b+d}{a+c}\right) \tag{5}$$

First, this property is true as we can easily prove by considering possible cases:

1. If $a \leq b$ and $c \leq d \Rightarrow \frac{a}{b} = \min\left(\frac{a}{b}, \frac{b}{a}\right), \frac{c}{d} = \min\left(\frac{c}{d}, \frac{d}{c}\right), \frac{a+c}{b+d} = \min\left(\frac{a+c}{b+d}, \frac{b+d}{a+c}\right)$. Without loss of generality $\frac{a}{b} = \min\left(\frac{a}{b}, \frac{c}{d}\right)$. Then, it is easy to see that $\frac{a}{b} \leq \frac{a+c}{b+d} \Leftrightarrow ab + ad \leq ab + bc \Leftrightarrow ad \leq bc \Leftrightarrow \frac{a}{b} \leq \frac{c}{d}$.
2. If $a \geq b$ and $c \geq d \Rightarrow \frac{b}{a} = \min\left(\frac{a}{b}, \frac{b}{a}\right), \frac{d}{c} = \min\left(\frac{c}{d}, \frac{d}{c}\right), \frac{b+d}{a+c} = \min\left(\frac{a+c}{b+d}, \frac{b+d}{a+c}\right)$. Then, the argument from the first case follows identically.
3. If $a \leq b$ and $c \geq d$, then we need to show that $\min\left(\frac{a}{b}, \frac{d}{c}\right) \leq \min\left(\frac{a+c}{b+d}, \frac{b+d}{a+c}\right)$. Without loss of generality, $\frac{a+c}{b+d} = \min\left(\frac{a+c}{b+d}, \frac{b+d}{a+c}\right)$. Now we have two cases:
   - If $\frac{a}{b} \leq \frac{d}{c}$, then $\frac{a}{b} \leq \frac{a+c}{b+d} \Leftrightarrow ab + ad \leq ab + ac \Leftrightarrow ad \leq bc \Leftrightarrow \frac{a}{b} \leq \frac{c}{d}$, which is true since $\frac{a}{b} \leq \frac{d}{c} \leq \frac{c}{d}$.
   - If $\frac{a}{b} \geq \frac{d}{c}$, then $\frac{d}{c} \leq \frac{a+c}{b+d} \Leftrightarrow bd + d^2 \leq ac + c^2$, which is true since $bd \leq ac \Leftrightarrow \frac{a}{b} \geq \frac{d}{c}$ and $d^2 \leq c^2 \Leftrightarrow d \leq c$.
4. If $a \geq b$ and $c \leq d$, then the proof follows identically to case 3.

Then, if equation 5 holds, then $\textsc{Balance}(\mathcal{C}) \leq \textsc{Balance}(\mathcal{C}')$. To see this, we there is a cluster index $i > 2$ for which $\textsc{Balance}(\mathcal{C}) = \textsc{Balance}(C_i)$. Then, $\textsc{Balance}(C_i) \leq \min\left(\min\left(\frac{a}{b}, \frac{b}{a}\right), \min\left(\frac{c}{d}, \frac{d}{c}\right)\right)$, so $\textsc{Balance}(\mathcal{C}') = \textsc{Balance}(C_i) \Rightarrow \textsc{Balance}(\mathcal{C}) = \textsc{Balance}(\mathcal{C}')$. If there is no such cluster $i > 2$, then $\textsc{Balance}(\mathcal{C}) = \min\left(\min\left(\frac{a}{b}, \frac{b}{a}\right), \min\left(\frac{c}{d}, \frac{d}{c}\right)\right) \leq \textsc{Balance}(C_i), \forall i > 2$. Then, it follows that $\textsc{Balance}(\mathcal{C}) \leq \textsc{Balance}(\mathcal{C}')$.

The Sum of Imbalances and Max Imbalance objectives are derived from the Balance objective. As they are only a function of the number of data points of different attributes in each cluster, they are also clearly pattern-based. To see that Sum of Imbalances is also mergeable, we consider again the base case of the induction proof, for 2 clusters. For the SUM OF IMBALANCES objective, note that $|C_i^1 - C_i^2| + |C_i^1 - C_i^2| \geq |C_i^1 - C_i^2 + C_i^1 - C_i^2|$ by the triangle inquality for any clusters C_i, C_j. In merging two clusters C_i and C_j from a clustering $\mathcal{C}$, obtaining a clustering $\mathcal{C}'$, the contribution to the objective of $C_i \cup C_j$ and the empty cluster is exactly $|C_i^1 - C_i^2 + C_i^1 - C_i^2|$ (an empty cluster has an imbalance of 0). Since all other clusters remained unchanged, we conclude that SUM OF IMBALANCES$(\mathcal{C}) \geq$ SUM OF IMBALANCES$(\mathcal{C}')$, and thus, merging two clusters can only improve the objective.

Remark B.2. As a note, the MAX IMBALANCE objective is not mergeable. As a simple objective, take the follwing clustering $\mathcal{C}$: each of the clusters C_1 and C_2 has one data point of attribute 1 and two data points of attribute 2. Then, MAX IMBALANCE$(\mathcal{C}) = 1$. The clustering $\mathcal{C}'$ obtained by merging clusters C_1 and C_2 has MAX IMBALANCE$(\mathcal{C}') = 2$.

Finally, for the induction step, if mergeability holds for merging a set of w clusters, then it holds for merging a set of $w + 1$ clusters as well, by reducing to the base case: without loss of generality, denote the $w + 1$ clusters to be merged by $C_1, \cdots, C_{w+1}$. By the induction hypothesis, mergeability holds for merging any w clusters, so for $\cup_{j\in[w]} C_j$. Then, the base case applies for the clusters $\cup_{j\in[w]} C_j$ and C_{w+1}.

Proportional violation objectives: First, we easily note that for any two clusterings $\mathcal{C}$ and $\mathcal{C}'$ that induce the same pattern Δ_a^C, we pair each cluster $C \in \mathcal{C}$ with a different cluster $C' \in \mathcal{C}'$ such that $|C| = |C'|$ and $|C^a| = |C'^a|$ for all attributes $a \in [l]$. Thus, from equation 2 it follows that $\Delta_a^C = \Delta_a^{C'}, \forall a \in [l]$. Since all proportional violation-based objectives defined are only a function of $(\Delta_a^C)_{a,C}$, it follows that all clusterings that have the same pattern will also have the same objective value for all four of the proportional violation-based objectives.

Finally, we will show that they are also mergeable. We show this again by induction over the number of merged clusters for each of the objectives. We start with the base case of two clusters, denoted without loss of generality by C_1 and C_2. We assume that neither C_1 and C_2 are empty. We say that C_1 and C_2 are part of a clustering $\mathcal{C}$, and we aim to show that the clustering $\mathcal{C}'$ in which C_1 and C_2 got merged will have OBJECTIVE$(\mathcal{C}') \leq$ OBJECTIVE$(\mathcal{C})$, for OBJECTIVE is one of the GROUP UTILITARIAN, GROUP UTILITARIAN-SUM, GROUP EGALITARIAN, GROUP EGALITARIAN-SUM objectives. For a sensitive attribute $a \in [l]$, assume without loss of generality that $\frac{|C_1^a|}{|C_1|} \leq \frac{|C_2^a|}{|C_2|}$. Then, the following property holds, also known as the *mediant inequality*:

$$\frac{|C_1^a|}{|C_1|} \leq \frac{|C_1^a| + |C_2^a|}{|C_1| + |C_2|} \leq \frac{|C_2^a|}{|C_2|} \tag{6}$$

To see this, note that the left hand side is equivalent to:

$$\begin{gathered}\frac{|C_1^a|}{|C_1|} \leq \frac{|C_1^a| + |C_2^a|}{|C_1| + |C_2|} \Leftrightarrow |C_1^a| \cdot |C_1| + |C_1^a| \cdot |C_2| \leq |C_1^a| \cdot |C_1| + |C_1| \cdot |C_2^a| \Leftrightarrow \\ |C_1^a| \cdot |C_2| \leq |C_1| \cdot |C_2^a| \Leftrightarrow \frac{|C_1^a|}{|C_1|} \leq \frac{|C_2^a|}{|C_2|}\end{gathered} \tag{7}$$

The right hand side is equivalent to:

$$\begin{gathered}\frac{|C_1^a| + |C_2^a|}{|C_1| + |C_2|} \leq \frac{|C_2^a|}{|C_2|} \Leftrightarrow |C_1^a| \cdot |C_2| + |C_2^a| \cdot |C_2| \leq |C_1| \cdot |C_2^a| + |C_2| \cdot |C_2^a| \Leftrightarrow \\ |C_1^a| \cdot |C_2| \leq |C_1| \cdot |C_2^a| \Leftrightarrow \frac{|C_1^a|}{|C_1|} \leq \frac{|C_2^a|}{|C_2|}\end{gathered} \tag{8}$$

By the definition from equation 2, we have

$$\begin{aligned}\beta_a - \Delta_a^{C_1} \leq \frac{|C_1^a|}{|C_1|} \leq \alpha_a + \Delta_a^{C_1}, \\ \beta_a - \Delta_a^{C_2} \leq \frac{|C_2^a|}{|C_2|} \leq \alpha_a + \Delta_a^{C_2},\end{aligned} \tag{9}$$

for clusters $C_1, C_2 \in \mathcal{C}$. As C_1 and C_2 got merged in $\mathcal{C}'$, they got replaced by the empty cluster $C_\emptyset$ and the merged cluster $C_1 \cup C_2$ in $\mathcal{C}'$. We note that $\Delta_a^{C_\emptyset} = \beta_a$, while $\Delta_a^{C_1 \cup C_2}$ is the minimum value that satisfies

$$\beta_a - \Delta_a^{C_1 \cup C_2} \leq \frac{|C_1^a| + |C_2^a|}{|C_1| + |C_2|} \leq \alpha_a + \Delta_a^{C_1 \cup C_2} \tag{10}$$

Using inequalities 9 and 10 with the mediant inequality, we get that

$$\begin{aligned} \beta_a - \Delta_a^{C_1} \leq \frac{|C_1^a| + |C_2^a|}{|C_1| + |C_2|} \leq \alpha_a + \Delta_a^{C_2} \Rightarrow \\ \beta_a - \max(\Delta_a^{C_1}, \Delta_a^{C_2}) \leq \frac{|C_1^a| + |C_2^a|}{|C_1| + |C_2|} \leq \alpha_a + \max(\Delta_a^{C_1}, \Delta_a^{C_2}) \end{aligned} \tag{11}$$

Since by definition, $\Delta_a^{C_1 \cup C_2}$ is the minimum value that satisfies equation 10, we get that

$$\Delta_a^{C_1 \cup C_2} \leq \max(\Delta_a^{C_1}, \Delta_a^{C_2}) \tag{12}$$

Note that the empty cluster will have $\Delta_a^{\emptyset} = 0, \forall a \in [l]$, so $\max(\Delta_a^{\emptyset}, \Delta_a^{C_1 \cup C_2}) = \Delta_a^{C_1 \cup C_2}$. This also implies that $\max\limits_{C \in \mathcal{C}} \Delta_a^C \geq \max\limits_{C \in \mathcal{C}'} \Delta_a^C$. Since the attribute $a \in [l]$ was chosen arbitrarily, we also get that

$$\sum_{a \in [l]} \max_{C \in \mathcal{C}} \Delta_a^C \geq \sum_{a \in [l]} \max_{C \in \mathcal{C}'} \Delta_a^C, \tag{13}$$

and thus the GROUP UTILITARIAN objective cannot increase by merging two clusters. Similarly, for the GROUP UTILITARIAN-SUM, since $0 = \Delta_a^{\emptyset} \leq \min(\Delta_a^{C_1}, \Delta_a^{C_2})$ and $\Delta_a^{C_1 \cup C_2} \leq \max(\Delta_a^{C_1}, \Delta_a^{C_2})$, and all other clusters have the same proportional violation in both clusterings, we also get that $\sum\limits_{C \in \mathcal{C}} \Delta_a^C \geq \sum\limits_{C \in \mathcal{C}'} \Delta_a^C$. Since the attribute $a \in [l]$ was chosen arbitrarily, we also get that

$$\sum_{a \in [l]} \sum_{C \in \mathcal{C}} \Delta_a^C \geq \sum_{a \in [l]} \sum_{C \in \mathcal{C}'} \Delta_a^C, \tag{14}$$

and thus the GROUP UTILITARIAN-SUM objective can also not increase by merging two clusters. Furthermore, equation 12 implies that $\max(\Delta_a^{\emptyset}, \Delta_a^{C_1 \cup C_2}) \leq \max(\Delta_a^{C_1}, \Delta_a^{C_2})$ which in turn implies that $\max\limits_{C \in \mathcal{C}} \Delta_a^C \geq \max\limits_{C \in \mathcal{C}'} \Delta_a^C$. Since this holds for any arbitrary $a \in [l]$, it also implies that

$$\max_{a \in [l], C \in \mathcal{C}} \Delta_a^C \geq \max_{a \in [l], C \in \mathcal{C}'} \Delta_a^C \tag{15}$$

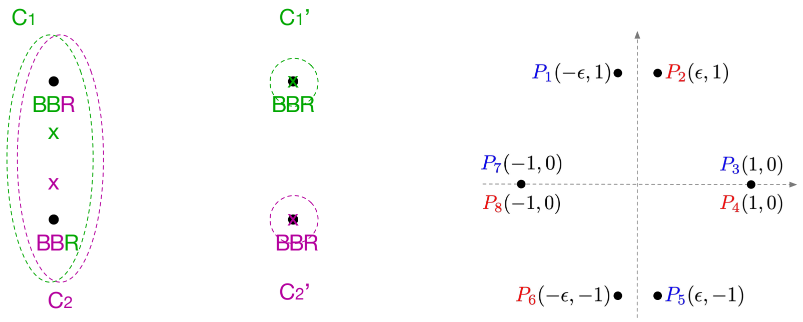
and thus the GROUP EGALITARIAN objective cannot increase by merging two clusters. Finally, since $0 = \Delta_a^{\emptyset} \leq \min(\Delta_a^{C_1}, \Delta_a^{C_2})$ and $\Delta_a^{C_1 \cup C_2} \leq \max(\Delta_a^{C_1}, \Delta_a^{C_2})$, and all other clusters have the same proportional violation in both clusterings, we also get that $\sum\limits_{C \in \mathcal{C}} \Delta_a^C \geq \sum\limits_{C \in \mathcal{C}'} \Delta_a^C$. Since the attribute $a \in [l]$ was chosen arbitrarily, we also get that

$$\max_{a \in [l]} \sum_{C \in \mathcal{C}} \Delta_a^C \geq \max_{a \in [l]} \sum_{C \in \mathcal{C}'} \Delta_a^C, \tag{16}$$

and thus the GROUP EGALITARIAN-SUM objective cannot increase by merging two clusters.

For the induction step, the proof is identical to the proof for the BALANCE objective: if mergeability holds for merging a set of w clusters, then it holds for merging a set of $w+1$ clusters as well, by reducing to the base case: without loss of generality, denote the $w+1$ clusters to be merged by $C_1, \cdots, C_{w+1}$. By the induction hypothesis, mergeability holds for merging any w clusters, so for $\cup_{j \in [w]} C_j$. Then, the base case applies for the clusters $\cup_{j \in [w]} C_j$ and C_{w+1} for all objectives.

□



(a) (b)

Figure 3: (a) An illustration of clustering under for non-pattern based fairness objectives. (b) An illustration of the $(P_i)_{i\in[8]}$ sets for non-mergeable fairness objectives.

B.2 A discussion on pattern-based and mergeability properties

Example of non-pattern based fairness objectives. As mentioned in the introduction, informally, a pattern-based fairness objective computes a per-cluster fairness quantity that only depends on the number of data points from each sensitive attribute. In practice, many fairness objectives satisfy this property, as they aim to operationalize different versions of *proportional representation*, with the goal of having balanced clusters among different sensitive attributes. Objectives that define fairness through the proportion of k-centers that are closest to them are not pattern-based, since they do not depend on the number of other nodes of different attributes in the same cluster but rather on equalizing the proportions of k-center assignments to different attributes [17, 41, 45] or the max average distance between points of different groups to their centers [31]. For example, the definition in Ghadiri et al. [31] states that for two groups R (red) and B (blue), the fairness objective is defined for a clustering $\mathcal{C}$ as

$$\Phi(S,\mathcal{C}) = \max\left(\frac{f(S,\mathcal{C}\cap\mathcal{X}^R)}{|R|}, \frac{f(S,\mathcal{C}\cap\mathcal{X}^B)}{|B|}\right), \tag{17}$$

where $\mathcal{X}^A$ is the subset of points from $\mathcal{X}$ with attribute A. Take the following set of points: 3 overlapping points x_1, x_2, x_3, with $x_1, x_2 \in B, x_3 \in R$, and 3 overlapping points x_4, x_5, x_6 at different coordinates than the first 3, with $x_4, x_5 \in B, x_6 \in R$. Then, the clustering $C_1 = \{x_1, x_2, x_6\}$, $C_2 = \{x_3, x_4, x_5\}$ has a non-zero fairness objective value with respect to its true centers (by computing the centroids). The clustering $C_1' = \{x_1, x_2, x_3\}$, $C_2' = \{x_4, x_5, x_6\}$, however, has the same pattern as (C_1, C_2), but a fairness objective value of 0 with respect to its true centroids. Even if we evalute (C_1, C_2) with respect to the centroids of (C_1', C_2'), it still has a non-zero fairness objective value. See Figure 3 (a) for an illustration.

Example of non-mergeable fairness objectives. As discussed in the introduction, a mergeable objective means that the objective value can only improve or stay the same when merging any number of clusters. While many fairness objectives are mergeable, we give an example below of non-mergeable objectives. We note that objectives that enforce a minimum number of data points in each cluster tend to not be mergeable. For example, the τ-ratio objective defined by [32] is non-mergeable, where the objective is defined as:

$$\sum_{x_i\in X} \mathbb{I}(x_i \in C_j)\mathbb{I}(\sigma(x_i){=}a){\geq}\tau_\ell \sum_{x_i\in X} \mathbb{I}(x_i \in C_j)\ \forall j \in [k] \text{ and } \forall a\ \in [l] \tag{18}$$

The τ-ratio enforces a minimum number of the total points that must go in each cluster, so empty clusters violate the mergeability condition on the fairness objective (since the number of clusters k is fixed).

Pareto front approximation gap gets arbitrarily large without mergeability. We rely on the assignment problem to provide an approximation for the clustering problem when computing the

Pareto front, which works for pattern-based and mergeable fairness objectives. However, for fairness objectives that are pattern-based but are **not** mergeable, if we apply directly Algorithm 1, without using a modified fairness function to filter the dominated patterns and adjust the clusterings, as we did in the proof of Theorem 3.5, then the resulting approximation ratio is no longer necessarily bounded, as we show in the example below. We showcase this for the τ-ratio objective.

We construct a dataset $\mathcal{X}$ such that $|\mathcal{X}| = 8m$ that contains 8 sets of m points each on the plane. Each point has a sensitive attribute, denote by b (blue) or r (red). We denote these sets by $(P_i)_{i\in[8]}$, constructed in the Euclidean space with the following coordinates (see Figure 3 (b) for an illustration):

- P_1 contains $2m-1$ blue points situated at coordinates $(-\epsilon, 1)$
- P_2 contains $2m-1$ red points situated at coordinates at $(\epsilon, 1)$
- P_3 contains 1 blue point situated at coordinates $(1, 0)$
- P_4 contains 1 red point of situated at coordinates $(1, 0)$
- P_5 contains $2m-1$ blue points situated at coordinates $(\epsilon, -1)$
- P_6 contains $2m-1$ red points situated at coordinates $(-\epsilon, -1)$
- P_7 contains 1 blue point situated at coordinates $(-1, 0)$
- P_8 contains 1 red point situated at coordinates $(-1, 0)$

Set $\epsilon < \frac{1}{8m}$, and $k = 4$. Then, the cost of not assigning the 4 points at $(1, 0)$ and $(-1, 0)$ their own two centers outweighs the benefits of assigning 2 centers to the points clustered near $(0, 1)$ or $(0, -1)$. Therefore, the best k-means and k-median clustering yields centers $S = \{(0, 1), (1, 0), (0, -1), (-1, 0)\}$.

We set $\tau = \frac{1}{4}$ as our fairness constraint, which constrains every center to have exactly m red points and exactly m blue points. Observe that the optimal quality way ϕ to assign such points to s_2 is to assign $m-1$ of the red points in P_2 and $m-1$ of the blue points in P_5 to S_2 (in addition to the two points assigned to s_2 by the unfair k-means/medians clustering). So the center at s_2 clusters together 2 sets of $m-1$ points separated by distance 2 on the plane. Even if we allow moving the location of s_2 after the assignment, the clustering cost is still lower bounded by$(2m-2)^{1/p}$, since the best center is unit distance from $2(m-1)$ points. Symmetrically, we do the corresponding assignment to s_4 and get a clustering cost lower bound of $2(2m-2)^{1/p}$.

Setting our centers to be $S^* = \{(0, 1), (0, 1), (0, -1), (0, -1)\}$, the assignment function ϕ^* with optimal quality under the fairness constraint maps m points from each of P_1 and P_2 to s_1^* and maps the remainder of the points from P_1 and P_2, as well as the points in P_3 and P_4 to s_2^*. Symmetrically, we do the corresponding assignment to s_3^* and s_4^* with the points from P_5, P_6, P_7, and P_8. This clustering achieves a clustering cost at most $((8m-4)\epsilon^p + 4(\sqrt{2})^p)^{1/p}$, where the first term comes from the $8m-4$ points in sets P_1, P_2, P_5, P_6 and the second term comes from the 4 points in sets P_3, P_4, P_7, P_8.

Let ϕ be the lowest cost assignment to centers S, and let S^m be the best possible centers for ϕ. Then we have that:

$$\frac{c(\phi, S^m)}{c(\phi^*, S^*)} \geq \frac{2(2m-2)^{1/p}}{((8m-4)\epsilon^p + 4(\sqrt{2})^p)^{1/p}} \tag{19}$$

Considering the k-means or the k-median clustering objectives, we get $(8m-4)\epsilon^p \leq 1$, since we set $\epsilon < \frac{1}{8m}$. Therefore,

$$\lim_{m\to\infty} \frac{c(\phi, S^m)}{c(\phi^*, S^*)} = \infty \tag{20}$$

Thus, for any constant b, the assignment Pareto set is not a $(b, 1)$ multiplicative approximation for the clustering Pareto set for non-mergeable fairness functions. This justifies the need for a modified algorithm for non-mergeable fairness objectives that can change the set of centers while searching for the best assignment, as described in Appendix A.

C Datasets details

For all datasets, we subsample $1,000$ data points, as the datasets sizes are prohibitively large. For the Adult and Census datasets, we use all numerical features available in the data and embed them in Euclidean space. For the BlueBike dataset, we use the starting and ending latitude and longitude as coordinates, directly embedded in Euclidean space. For all datasets, the gender of users is self-reported. Table 1 describes the characteristics of all datasets.

Table 1: Data and experimental details.

	# of features	Sensitive attribute	δ
Adult	5	Gender	0.05
Census1990	66	Gender	0.001
BlueBike	4	Gender	0.01

We note that some datasets naturally cluster into fairer clusters than others. For this reason, setting δ too high renders a single point on the Pareto front, since the vanilla clustering itself will be fair for proportional violation-based objectives. In fact, the proportional violation value will be equal to 0. For this reason, we experiment with various values of δ, reported in Table 1 for the experiments presented in the main text.

D Repeated-FCBC: Experimental Details

As described in the main text, we investigate alternatives for faster recovery of the Pareto front in the case of two objectives: clustering cost and fairness. We employ a recently proposed linear programming method that bounds the clustering objective and optimizes a fairness objective, up to an approximation, denoted the FCBC algorithm (Algorithm 1 in Esmaeili et al. [26]). We implement the FCBC algorithm repeatedly by discretizing over the clustering cost space. We perform a sweep over the upper bound U values, setting its minimum value equal to the clustering cost of a vanilla clustering algorithm and its max value equal to a constant times the clustering cost of a vanilla clustering algorithm, for some chosen constant. We describe the approach formally in Algorithm 2, with an illustration in Figure 4.

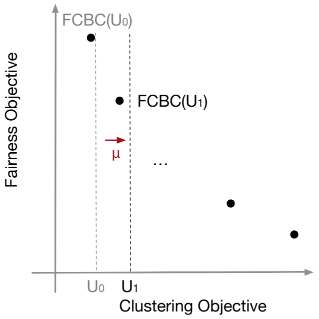


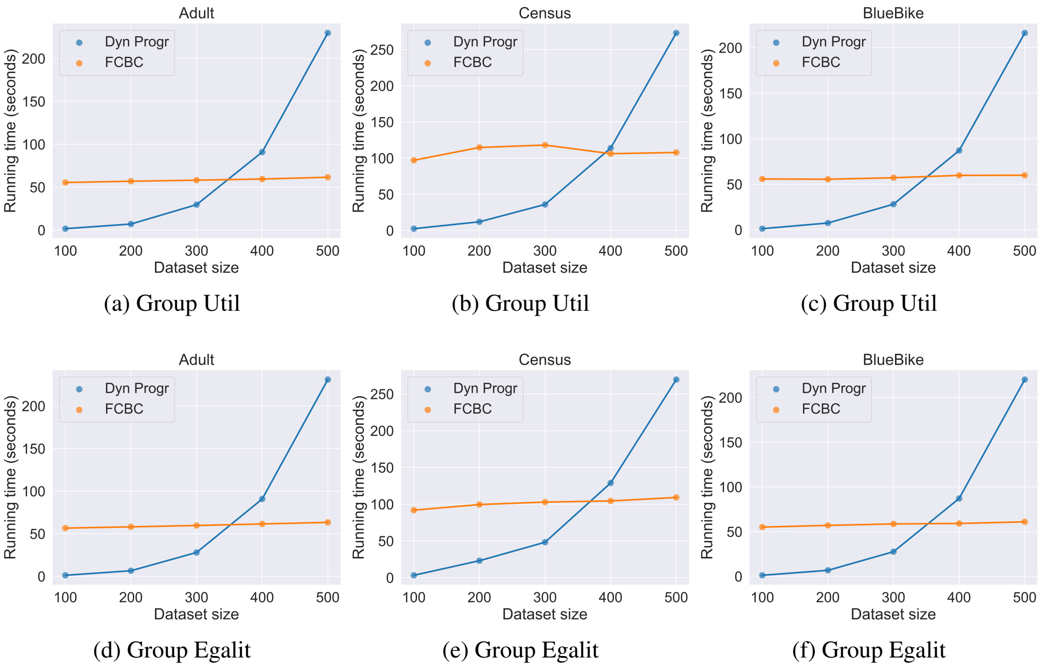
Figure 4: An illustration of implementing the repeated FCBC algorithm as the clustering cost upper bound U varies.

We compare the Pareto front recovered from Algorithm 2 with the Pareto front recovered from the dynamic programming approach in Algorithm 1 on all the real-world datasets in Figure 2 in Section 4, setting $U_{\max} = U_{\min} \cdot C$, where $U_{\min}$ is the clustering cost of a vanilla clustering algorithm (in our case, k-means++), $C = 1.5$, and $N = 50$. We note that setting C equal to 1 is equivalent to constraining the clustering cost in the FCBC algorithm to be equal to the vanilla clustering algorithm cost.

Running time comparison. We illustrate in Figure 5 the running time comparison between the dynamic programming approach from Algorithm 1, labeled as 'Dyn Progr', and the repeated FCBC approach from Algorithm 2, labeled as 'FCBC', for samples of different sizes of all the datasets. We showcase two objectives, GROUP UTILITARIAN and GROUP EGALITARIAN, since the original FCBC algorithm [26] is designed only for these two, among the five objectives we described in Section 4. We note that Algorithm 1 has a similar running time for all other objectives. The experimental running time follows the analysis presented in Section 3: for $k = 2$ clusters and $l = 2$ sensitive attributes, one would expect a running time of $O(n^2)$.

Algorithm 2 Repeated FCBC for recovering the Pareto Front

```
Input: $U_{\min}, U_{\max}, N, \textsc{Clustering-Obj}, \textsc{Fairness-Obj}$.
Step 1: Discretize the interval $[U_{\min}, U_{\max}]$: set $\eta = (U_{\max} - U_{\max})/N$ and
for $i \leftarrow 0$ to $N$ do
    Set $U_i = U_{\min} + \eta \cdot i$
Step 2: apply the FCBC algorithm for the discretized clustering cost bound sequence $(U_i)_i$, adding results to the recovered Pareto front approximation $X_P^{\mathrm{FCBC}}$, initialized to $X_P^{\mathrm{FCBC}} = \emptyset$:
for $i \leftarrow 0$ to $N$ do
    $X_P^{\mathrm{FCBC}} \leftarrow X_P^{\mathrm{FCBC}} \cup \{FCBC(U_i, \textsc{Clustering-Obj}, \textsc{Fairness-Obj})\}$
Output: $X_P^{\mathrm{FCBC}}$
```



(a) Group Util (b) Group Util (c) Group Util

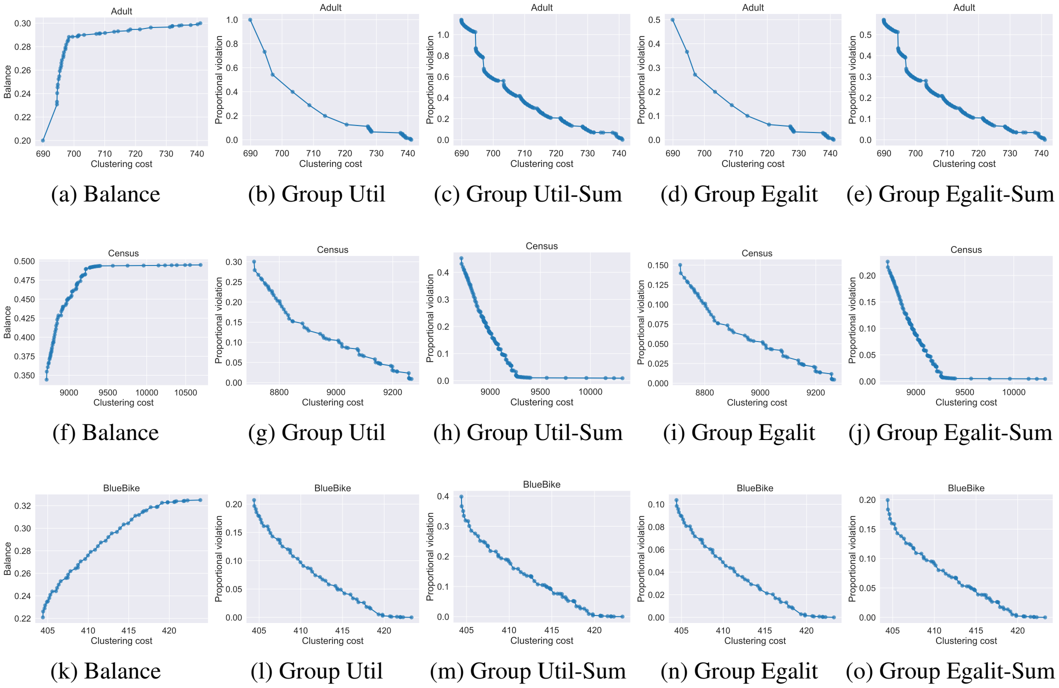
(d) Group Egalit (e) Group Egalit (f) Group Egalit

Figure 5: Running time comparison with our dynamic programming approach from Algorithm 1, labeled as 'Dyn Progr', and the repeated FCBC approach from Algorithm 2, labeled as 'FCBC', for each dataset (by column) and for the GROUP UTILITARIAN and GROUP EGALITARIAN objective (by row).

E Additional Experiments

We present experimental results on the three datasets and the five fairness objectives defined in Section 4 for $k = 3$ clusters in Figures 6 and 7. We note that results are qualitatively similar as for $k = 2$ clusters. As mentioned in the main text, we note that the Pareto front need not be strictly convex for two minimization objectives, nor strictly concave for a minimization objective and a maximization objective (as in the case of the clustering objective and the balance objective), since it simply consists of the undominated points.

For the interested reader, we also showcase the Pareto front for the SUM OF IMBALANCES objective, for all three datasets, for $k = 2$ and $k = 3$, in Figure 8. We note that since this objective is best suited for data with relatively equal proportions between the two groups, we subsample equal proportions of each gender for each of the three datasets, Adult, Census, and BlueBike.

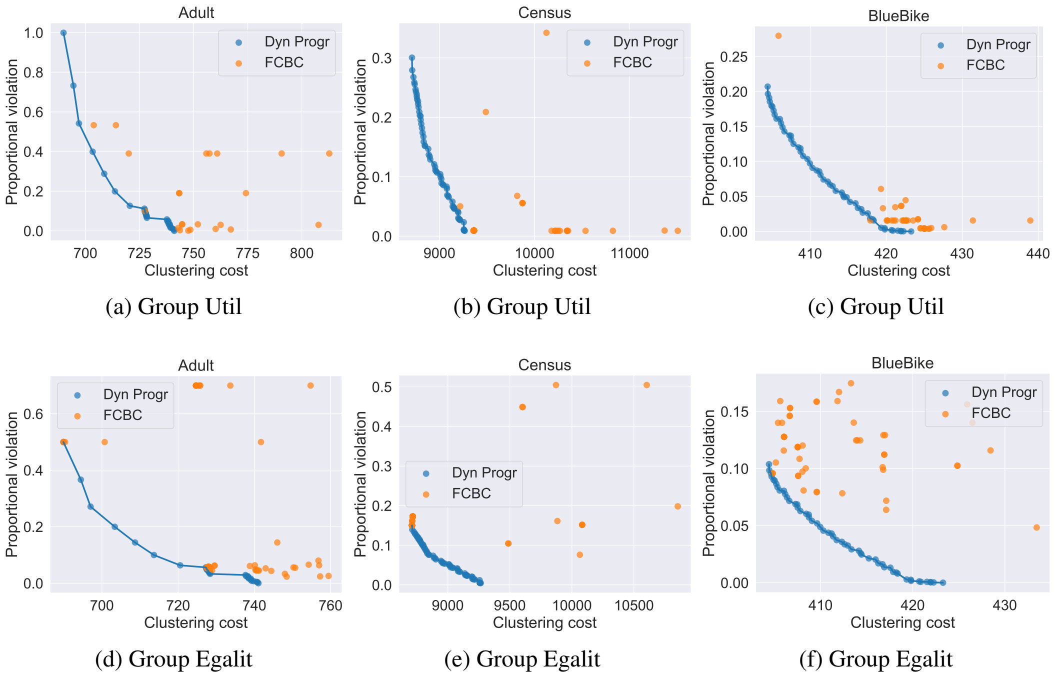


(a) Balance (b) Group Util (c) Group Util-Sum (d) Group Egalit (e) Group Egalit-Sum

(f) Balance (g) Group Util (h) Group Util-Sum (i) Group Egalit (j) Group Egalit-Sum

(k) Balance (l) Group Util (m) Group Util-Sum (n) Group Egalit (o) Group Egalit-Sum

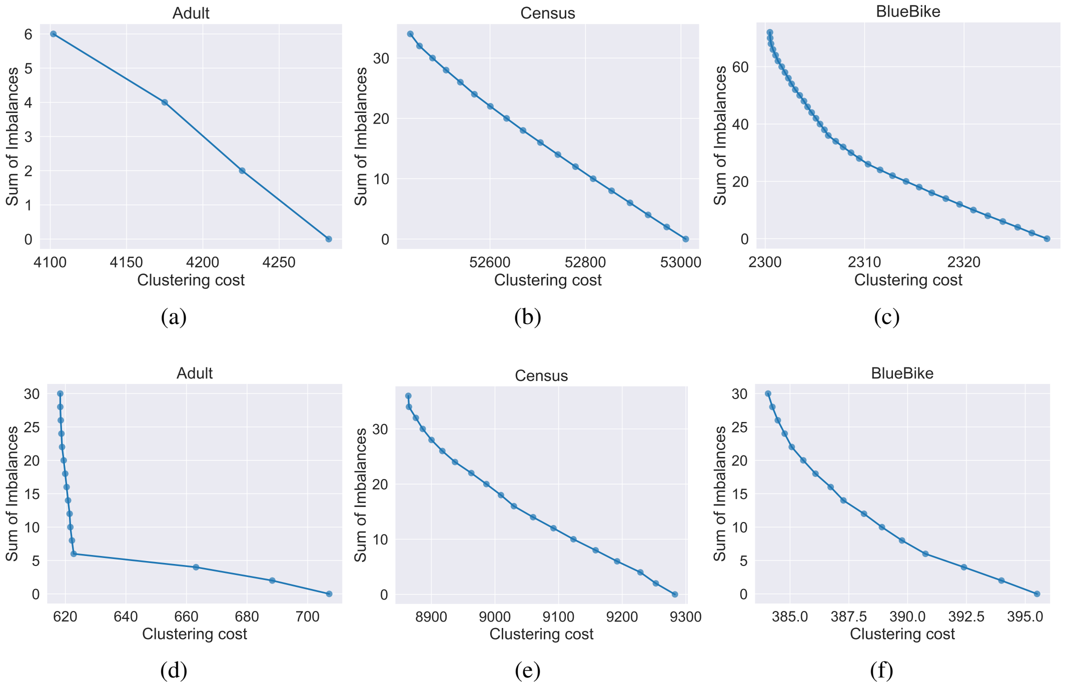
Figure 6: Pareto front recovered by Algorithm 1 for the Adult, Census, and BlueBike datasets (by row), for various fairness objectives (by column), for $k = 3$ clusters.



(a) Group Util (b) Group Util (c) Group Util

(d) Group Egalit (e) Group Egalit (f) Group Egalit

Figure 7: Pareto front recovered by Algorithm 1 (labeled as 'Dyn Progr', in blue) and by Algorithm 2 (labeled as 'FCBC', in orange) for the Adult, Census, and BlueBike datasets (by column) and for the Group Utilitarian and Group Egalitarian objectives (by row), for $k = 3$ clusters.



(a) (b) (c)

(d) (e) (f)

Figure 8: Pareto front recovered for the SUM OF IMBALANCES objective for the Adult, Census, and BlueBike datasets (by column), for $k = 2$ (top row) and $k = 3$ (bottom row) clusters.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We provide a clear summary of our results in the abstract and introduction, stating all conditions needed for the results to hold. We summarize the experiments presented in the paper, with details in the main section and the Appendix.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses all the necessary conditions for the results to hold. In particular, all our algorithms require that the first objective (the clustering objective) is based on a metric and that the second objective (the fairness objective) is pattern-based. Algorithm 1 also requires that the fairness objective is mergeable. We give theoretical bounds for the running time of all algorithms and we showcase the running time empirically for the algorithms present in the experimental section. We clearly state the datasets and parameters used. We discuss limitations in the discussion section in detail.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All our theoretical results clearly state the necessary conditions in the body of the theorems (e.g., Theorems 3.2–3.6) require a clustering objective based on a metric and a pattern-based fairness objective, while Theorem 3.4 also requires a mergeable fairness objective. The main paper contains sketches of all proofs and the Appendix contains all the detailed proofs.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the pseudocode of all algorithms used in the paper and the Appendix, with detailed parameters choices, instantiation details, and number of runs. All the code and data is available at this link.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All the code and data is available at this link. All experimental details are provided in the main paper as well as in the Appendix for reproducibility of results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide all experimental details (choices of input, parameter settings, data details, and number of runs) in the main paper as well as the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The results do not require error bars, as there is always the same set of Pareto points recovered for the Pareto front for the assignment problem when using a standard vanilla k-means clustering algorithm (in our case, k-means++) as an input to our algorithms. We subsample the data once and regarded fixed from there on.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details on the machines used to run the experiments, the libraries needed, as well as an empirical running time analysis in the paper, both in the Experiments section as well as in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper conforms to the NeurIPS Code of Ethics. In particular, the datasets used are open widely used in related work; we do not use human participants; we discuss societal implications of our method with the purpose of improving fairness in clustering.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impact of our paper, in particular potential use cases by practitioners who wish to use our method for decision-making in choosing an optimal trade-off between clustering and fairness objectives, in cases where the data points represent people.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite all the datasets used, noting that they are open for use. We cite any code used for the implementation of Algorithm 2 properly. All other code is written and owned by the authors.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We describe all datasets used, with proper documentation of parameters used, algorithms, and code instructions. All datasets and code are made available at this link.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.