Fair Labeled Clustering

Seyed A. Esmaeili
University of Maryland, College Park
Maryland, USA
esmaeili@cs.umd.edu

Sharmila Duppala
University of Maryland, College Park
Maryland, USA
sduppala@cs.umd.edu

John P. Dickerson
University of Maryland, College Park
Maryland, USA
johnd@umd.edu

Brian Brubach
Wellesley College
Massachusetts, USA
bb100@wellesley.edu

ABSTRACT

The widespread use of machine learning algorithms in settings that directly affect human lives has instigated significant interest in designing variants of these algorithms that are provably fair. Recent work in this direction has produced numerous algorithms for the fundamental problem of clustering under many different notions of fairness. Perhaps the most common family of notions currently studied is group fairness, in which proportional group representation is ensured in every cluster. We extend this direction by considering the downstream application of clustering and how group fairness should be ensured for such a setting. Specifically, we consider a common setting in which a decision-maker runs a clustering algorithm, inspects the center of each cluster, and decides an appropriate outcome (label) for its corresponding cluster. In hiring for example, there could be two outcomes, positive (hire) or negative (reject), and each cluster would be assigned one of these two outcomes. To ensure group fairness in such a setting, we would desire proportional group representation in every label but not necessarily in every cluster as is done in group fair clustering. We provide algorithms for such problems and show that in contrast to their NP-hard counterparts in group fair clustering, they permit efficient solutions. We also consider a well-motivated alternative setting where the decision-maker is free to assign labels to the clusters regardless of the centers' positions in the metric space. We show that this setting exhibits interesting transitions from computationally hard to easy according to additional constraints on the problem. Moreover, when the constraint parameters take on natural values we show a randomized algorithm for this setting that always achieves an optimal clustering and satisfies the fairness constraints in expectation. Finally, we run experiments on real world datasets that validate the effectiveness of our algorithms.

KEYWORDS

Algorithmic Fairness, Unsupervised Learning, Clustering

ACM Reference Format:
Seyed A. Esmaeili, Sharmila Duppala, John P. Dickerson, and Brian Brubach. 2022. Fair Labeled Clustering. In *KDD '22, Aug 14–18, 2021, Washington DC Convention Center*. ACM, New York, NY, USA, 12 pages.

KDD '22, Aug 14–18, 2022, Washington DC Convention Center

1 INTRODUCTION

Machine learning applications have seen widespread use across diverse areas from criminal justice to hiring to healthcare. These applications significantly affect human lives and risk contributing to discrimination [4, 29]. As a result, research has been directed toward the creation of fair machine learning algorithms [17]. Much existing work has focused on the supervised setting. However, significant attention has recently been given to clustering—a fundamental problem in unsupervised learning and operations research. While many important notions of fair clustering have been proposed, the most relevant to our work is group (demographic) fairness [3, 6, 8, 9, 12, 15, 19, 26, 27]. In many of those works, fairness is maintained at the cluster level by imposing constraints on the proportions of groups present in each cluster. For example, we may require the racial demographics of each cluster to be close to the dataset as a whole (demographic/statistical parity) or that no group is over-represented in any cluster.

While constraining the demographics of each cluster is appropriate in some settings, it may be unnecessary or impractical in others. In decision making applications, each cluster eventually has a specific label (outcome) associated with it which may be more positive or negative than others. If the same label is applied to multiple clusters, we may only wish to bound the demographics of points associated with a given label as opposed to bounding the demographics of each cluster.

To be more concrete, consider the application of clustering for market segmentation in order to generate better targeted advertising [2, 11, 24, 34]. In this setting, we select or engineer features which are informative for targeted advertising and apply clustering (e.g., k-means) to the dataset. Then, we analyze the resulting centers (prototypical examples) and make decisions for targeted advertising in the form of recommending specific products or offering certain deals. These products or deals may have different levels of quality, i.e., we may assign labels such as: *mediocre*, *good*, or *excellent* to each cluster based on the quality of its advertisements. For the clusters of a given label (treated as one), it is possible that a certain demographic would be under-represented in the *excellent* label or that another could be over-represented in the *mediocre* label. In fact, the reports in [14, 28, 33] indicate that targeted advertising may under-represent certain demographics for some advertisements. An algorithm that ensures each group is represented proportionally in each label could remedy this issue. While applying group fair clustering algorithms would also ensure demographic representation

in the clusters and thus the labels, it could come at the price of a higher deformation in the clustering since points would have to be routed to possibly faraway centers just to satisfy the representation proportions. On the other hand, ensuring fair representation across the labels, but not necessarily the centers is less restrictive and likely to cause less deformation to the clustering.

Another similar example is clustering for job screening [30] in which we have a dataset of candidates,[1] and each candidate is represented as a point in a metric space. Clustering could be applied over this set to obtain k many clusters. Then, the center of each cluster is given a more costly examination (e.g., a human carefully screening a job application). Accordingly, the centers would be assigned labels from the set: *hire*, *short-list*, *scrutinize further*, or *reject*. Naturally, more than one cluster could be assigned the same label. Clearly, the greater concern here is demographic parity across the labels, but not necessarily the individual clusters. Thus, group fair clustering would yield unnecessarily sub-optimal solutions.

While in the above examples the label of the center was decided according to its position in the metric space. One can envision applications in Operations Research where the label assignment of the center is not dependent on its position [32, 36]. Rather, we would have a set of centers (facilities) of different service types (or quality) and we would have a budget for each service type. Further, to ensure group fairness we would satisfy the demographic representation over the service types offered. In this setting, we would have to choose the labels so as to minimize the clustering cost subject to further constraints such as budget and fair demographic representation.

The above examples illustrate the need for a group fairness definition at the label level when clustering is applied in decision-making settings or when the different centers (facilities) provide different types of services. In addition to being sufficient, evaluating fairness at the label level rather than cluster level can also be necessary. When the metric space is correlated with group membership it may be costly, counterproductive, or impossible to get meaningful clusters that each preserve the demographics of the dataset. For example, if the metric space is geographic as in many facility location problems, a person's location can be correlated with their racial group membership due to housing segregation. The same is true in machine learning when common features like location redundantly encode sensitive features such as race. In this case, the more strict approach of group fairness in each cluster could cause a large enough degradation in clustering quality that the entity in charge chooses a classical "unfair" clustering algorithm instead. In legal terms, this unfair clustering approach may exhibit *disparate impact*—members of a protected class may be adversely affected without provable intent on the part of the algorithm. However, disparate impact is allowed if the unfair clustering can be justified by *business necessity* (e.g., the fair clustering alternative is too costly)[35].

Thus, our work can be seen as a less stringent, less costly, and fundamentally different approach which still satisfies some similar fairness criteria to existing group fair clustering formulations. In addition, the decision-maker may not be concerned with the demographic representation in all labels, but rather only a specific set of label(s) such as *hire* and *short-list*. It may also be desired to enforce different lower and upper representation bounds for different labels.

[1] In some countries, such as India, the number of candidates can be in the millions for government jobs: https://www.bbc.com/news/world-asia-india-43551719.

1.1 Our Contributions

We introduce the problem of fairness in labeled clustering in which group fairness is ensured within the labels as opposed to each cluster. Specifically, we are given a set of centers found by a clustering algorithm, then having found the centers, we have to satisfy group fairness over the labels. We consider two settings: (1) **labeled clustering with assigned labels (**LCAL**)** where the center labels are decided based on their position as would be expected in machine learning applications and (2) **labeled clustering with unassigned labels (**LCUL**)** where we are free to select the center labels subject to some constraints. We note that throughout we consider the set of centers to be given and fixed (although in the unassigned setting their labels are unknown), therefore the problem is essentially a routing (assignment) problem where points are assigned to centers rather than a clustering problem. We however, refer to it as clustering since we minimize the clustering cost throughout and since our motivation is clustering based. Moreover, many of the application cases of the assigned labels setting would not alter the centers as that would not change the assigned labels which are given manually through further inspection [11, 30, 34] or in the case of the unassigned labels we would have a fixed set of centers. Further, the work of [15] in fair clustering follows a similar setting where the centers are fixed.

For the LCAL (assigned labels) setting, we show that if the number of labels is constant, then we can obtain an optimal clustering cost subject to satisfying fairness within labels in polynomial time. This is in contrast to the equivalent *fair assignment* problem in fair clustering which is NP-hard[9, 18].[2] Furthermore, for the important special case of two labels, we obtain a faster algorithm with running time $O(n(\log n + k))$.

For the LCUL (unassigned labels) setting, we give a detailed characterization of the hardness under different constraints and show that the problem could be NP-hard or solvable in polynomial time. Furthermore, for a natural specific form of constraints we show a randomized algorithm that always achieves an optimal clustering and satisfies the fairness constraints in expectation.

We conduct experiments on real world datasets that show the effectiveness of our algorithms. In particular, we show that our algorithms provide fairness at a lower cost than fair clustering and that they indeed scale to large datasets. We note that due to the space limit, some proofs are relegated to the appendix.

2 RELATED WORK

Much of the investigation into fairness in machine learning and automated systems was sparked by the seminal work of [17]. That work and others [20, 37] respond to the reality that points which should receive similar classifications, but belong to different demographic groups may not be near each other in the feature space.

[2] In this equivalent problem, the set of centers is given. We seek an assignment of points to these centers that minimizes a clustering objective and bounds the group proportions assigned to each center.

Our approach accounts for this phenomenon as well by allowing points from different groups to be distant in the metric space and assigned to different clusters, but receive the same label.

The most closely related work in the clustering space addresses group (demographic) fairness among the members of each cluster [3, 6, 8, 9, 12, 15, 18, 19, 26]. However, as noted earlier, these approaches can diverge quite a bit from the problem we consider and are not directly comparable. Some work also considers the less related fair data summarization problem of bounding group proportions among the set of centers/exemplars [27]. In addition, several other notions of fair clustering and summarization exist to capture the diverse settings and objectives for which fairness is desirable. These include service guarantees bounding the distance of points to centers [25], preserving nearby pairs or communities of points in the metric space [10], equitable group representation [1, 23], and fair candidate selection[7].

In particular, the setting of [15] is very similar to ours in that the set of centers is fixed, and the problem amounts to routing points to centers so as to minimize the clustering cost function. However, unlike our work, the constraint is to satisfy conventional group fairness in the clusters; whereas in our setting, we are concerned with group fairness only within the labels.

3 PRELIMINARIES AND PROBLEM FORMULATION

We are given a complete metric graph with a set of vertices (points) C where $|C| = n$. Further, each point has a color assigned to it according to the function $\chi : C \to \mathcal{H}$ where $\mathcal{H}$ is the set of possible colors, with cardinality R, i.e. $|\mathcal{H}| = R$. We refer to the set of points with color $h \in \mathcal{H}$ by C^h. We further have a distance function $d : C \times C \to \mathbb{R}_{\geq 0}$ which defines a metric. We are given a set S of centers that have been selected, S contains at most k many centers, i.e. $|S| \leq k$. Furthermore, we have the set of labels $\mathcal{L}$ where $\mathcal{L}$ has a total of m many possible labels, i.e. $|\mathcal{L}| = m$. The function $\ell : S \to \mathcal{L}$ assigns centers to labels. Our problem always involves finding an assignment from points to centers, $\phi : C \to S$ such that it is the optimal solution to a constrained optimization problem where the objective is a clustering objective. Specifically, we always have to minimize the objectives: $\Big(\sum_{j \in C} d^p(j, \phi(j))\Big)^{1/p}$, where $p = \infty, 1$, and 2 for the k-center, k-median, and k-means objectives, respectively. We note that for the k-center with $p = \infty$, the objective reduces to a simpler form $\Big(\sum_{j \in C} d^p(j, \phi(j))\Big)^{1/p} = \max_{j \in C} d(j, \phi(j))$ which is the maximum distance between a point j and its assigned center $\phi(j)$. We consider the number of colors R to be a constant throughout. This is justified by the fact that in most applications demographic groups tend to be limited in number.

As mentioned earlier, we have two settings and accordingly two variants of this optimization: (1) labeled clustering with assigned labels (LCAL) where the centers have already been assigned labels and (2) labeled clustering with unassigned labels (LCUL) where the centers have not been assigned any labels and can be assigned any arbitrary labels from the set $\mathcal{L}$ subject to (possible) additional constraints.

We pay special attention to the two label case where $\mathcal{L} = \{P, N\}$ with P being a positive outcome label and N being a negative outcome label, although many of our results can be extended to the general case where $|\mathcal{L}| = m > 2$.

3.1 Labeled Clustering with Assigned Labels (LCAL):

In this problem the labels of the centers have been assigned, i.e. the function ℓ is fully known and fixed. We look for an assignment ϕ which is the optimal solution to the following problem:

$$\min_{\phi} \Big(\sum_{j \in C} d^p(j, \phi(j))\Big)^{1/p} \tag{1a}$$

$$\forall L \in \mathcal{L}, \forall h \in \mathcal{H} : l_h^L \sum_{\substack{i \in S \\ \ell(i) = L}} |C_i| \leq \sum_{\substack{i \in S \\ \ell(i) = L}} |C_i^h| \leq u_h^L \sum_{\substack{i \in S \\ \ell(i) = L}} |C_i| \tag{1b}$$

$$\forall L \in \mathcal{L} : (LB)_L \leq \sum_{i \in S : \ell(i) = L} |C_i| \leq (UB)_L \tag{1c}$$

where C_i refers to the points ϕ assigns to the center i, i.e. $C_i = \{j \in C \,|\, \phi(j) = i\}$. $C_i^h = C_i \cap C^h$, i.e. the subset of C_i with color h. l_h^L and u_h^L are lower and upper proportional bounds for color h. Clearly, $l_h^L, u_h^L \in [0, 1]$. Constraints (1b) are the proportionality (fairness) constraints that are to be satisfied in fair labeled clustering. Notice how we have a superscript L in l_h^L and u_h^L, this is to indicate that we may desire different proportional representations in different labels. For example, for the case of two labels $\mathcal{L} = \{P, N\}$, we may not want to enforce proportional representation in the negative label so we set $l_h^N = 0$ and $u_h^N = 1$ but we may want to enforce lower representation bounds in the positive label and therefore set l_h^P to some non-trivial value. Note that these constraints generalize those of fair clustering, in fact we can obtain the constraints of fair clustering by letting each center have its own label ($m = k$) and enforcing the proportional representation bounds to be the same throughout all labels. However, in our problem we focus on the case where the number of labels m is constant since in most applications we expect a small number of labels (outcomes). In fact, a large number could cause a problem in terms of decision making and result interpretability.

In constraints (1c), $(LB)_L$ and $(UB)_L$ are pre-set upper and lower bounds on the number of points assigned to a given label, clearly $(LB)_L, (UB)_L \in \{0, 1, \ldots, n\}$. They are additional constraints we introduce to the problem that have not been previously considered in fair clustering. Our motivation comes from the fact that since positive or negative outcomes could be associated with different labels, it is reasonable to set an upper bound on the total number of points assigned to a positive label, since a positive assignment may incur a cost and there is a bound on the budget. Similarly, we may set a lower bound to avoid trivial solutions where most points are assigned to negative outcomes and no or very few agents enjoy the positive outcome.

3.2 Labeled Clustering with Unassigned Labels (LCUL):

In labeled clustering with unassigned labels LCUL, the labels of the centers have not been assigned. As noted, this captures certain OR applications in which the label of a center is not related to its position in the metric space.

Similar to the case with assigned labels LCAL, we would also wish to minimize the clustering objective. In general we have the following optimization problem:

$$\min_{\phi,\ell} \Big(\sum_{j \in C} d^p(j, \phi(j)) \Big)^{1/p} \tag{2a}$$

$$\forall L \in \mathcal{L}, \forall h \in \mathcal{H} : l_h^L \sum_{\substack{i \in S \\ \ell(i)=L}} |\, C_i \,| \leq \sum_{\substack{i \in S \\ \ell(i)=L}} |\, C_i^h \,| \leq u_h^L \sum_{\substack{i \in S \\ \ell(i)=L}} |\, C_i \,| \tag{2b}$$

$$\forall L \in \mathcal{L} : (LB)_L \leq \sum_{i \in S : \ell(i)=L} |\, C_i \,| \leq (UB)_L \tag{2c}$$

$$\forall L \in \mathcal{L} : (CL)_L \leq |S^L| \leq (CU)_L \tag{2d}$$

Note how in the above objective ℓ has been added as an optimization variable unlike the objective in (1) for LCAL. Further, we have added constraint (2d) where S^L refers to the subset of centers that have been assigned label L by the function ℓ, i.e. $S^L = \{i \in S | \ell(i) = L\}$. This constraint simply lower bounds S^L by $(CL)_L$ and upper bounds it by $(CU)_L$. This constraint models minimal service guarantees (lower bound) and budget (upper bound) guarantees. Clearly, $(CL)_L, (CU)_L \in \{0, 1, \ldots, k\}$. Further, setting $(CL)_L = 0$ and $(CU)_L = k \; \forall L \in \mathcal{L}$ allows any label to have any number of centers, effectively nullifying the constraint. We show in a subsequent section that forcing certain constraints on the problem can make it NP-hard and that relaxing some constraints would make the problem permit polynomial time solutions.

4 ALGORITHMS AND THEORETICAL GUARANTEES FOR LCAL

4.1 LCAL is Polynomial Time Solvable:

LCAL is problem (1) where we have a collection of centers and we wish to minimize a clustering objective subject to proportionality constraints (1b) and possible constraints on the number of points each label is assigned (1c). Fair allocation[3] is a problem which has a very similar form to our problem; the centers have already been decided and we wish to satisfy the same proportionality constraints in every cluster, specifically the optimization problem is:

$$\min_{\phi} \Big(\sum_{j \in C} d^p(j, \phi(j)) \Big)^{1/p} \tag{3a}$$

$$\forall i \in S, \forall h \in \mathcal{H} : l_h |\, C_i \,| \leq |\, C_i^h \,| \leq u_h |\, C_i \,| \tag{3b}$$

It may be thought that the above optimization is simpler than that of *LCAL* (1), since all clusters have to satisfy the same proportionality bounds and there is no bound on the total number of points assigned to a any specific cluster. However, [9, 18] show that the problem is in fact NP-hard for all clustering objectives. We show in the theorem below that *LCAL* can be solved in polynomial time for all clustering objectives.

Theorem 1. *Labeled clustering with assigned labels LCAL is solvable in polynomial time for the all clustering objectives (k-center, k-median, and k-means).*

Proof. The key observation is that any assignment function ϕ, will assign a specific number of points n_L to the centers with label L. Further, we have that $\sum_{L \in \mathcal{L}} n_L = n$ since all points must be covered. Now, since $|\, \mathcal{L} \,| = m$ is a constant, this means that there is a polynomial number of ways to vary the total number of points distributed across the labels. More specifically, the total number of ways to distribute points across the given labels is upper bounded by $\underbrace{n \times n \times \cdots \times n}_{m-1} = n^{m-1}$. Note that once we decide the number of points assigned to the first $(m-1)$ labels, the last label must be assigned the remaining amount to cover all n points, so we have a total of n^{m-1} possibilities. Since we have established, that there is a polynomial number of possibilities for distributing the number of points across the labels, if we can solve *LCAL* optimally for each possibility and simply take the minimum across all possibilities then we would obtain the optimal solution.

Now that we are given a specific distribution of number of points across labels, i.e. $(n_1, \ldots, n_L, \ldots, n_m)$ where $\sum_{L \in \mathcal{L}} n_L = n$, we have to solve *LCAL* optimally for that distribution. The problem amounts to routing points to appropriate centers such that we minimize the clustering objective and satisfy the distribution of number of points across the labels along with the color proportionality. To do that we construct a network flow graph and solve the resulting minimum cost max flow problem. The network flow graph is constructed as follows:

- **Vertices:** the set of vertices is $V = \{s\} \cup C \cup (\cup_{h \in \mathcal{H}} S^h) \cup (\cup_{h \in \mathcal{H}} \mathcal{L}^h) \cup \mathcal{L} \cup \{t\}$. Vertex s is the source, further we have a vertex for each point, hence the set of vertices C. For each color $h \in \mathcal{H}$ we create a vertex for each center in S and for each label in $\mathcal{L}$, these vertices constitute the sets $\cup_{h \in \mathcal{H}} S^h$ and $\cup_{h \in \mathcal{H}} \mathcal{L}^h$, respectively. We also have a vertex for each label in $\mathcal{L}$ and finally the sink t.
- **Edges:** the set of edges is $E = E_{s \to C} \cup E_{C \to S^h} \cup E_{S^h \to \mathcal{L}^h} \cup E_{\mathcal{L}^h \to \mathcal{L}} \cup E_{\mathcal{L} \to t}$. $E_{s \to C}$ consists of edges from the source s to every point $j \in C$, $E_{C \to S^h}$ consists of edges from every point $j \in C$ to the center of vertices of the same color in S^h, $E_{S^h \to \mathcal{L}^h}$ consists of edges from the colored centers to their corresponding label of the same color, $E_{\mathcal{L}^h \to \mathcal{L}}$ consists of edges from the colored labels to their corresponding label, finally $E_{\mathcal{L} \to t}$ consists of edges from every label in $\mathcal{L}$ to the sink t.
- **Capacities:** the edges of $E_{s \to C}$ have a capacity of 1, the edges of $E_{\mathcal{L}^h \to \mathcal{L}}$ have a capacity of $\left\lfloor u_h^L n_L \right\rfloor$, the edges of $E_{\mathcal{L} \to t}$ have a capacity of n_L.
- **Demands:** the vertices of $\mathcal{L}^h$ have a demand of $\left\lceil l_h^L n_L \right\rceil$, the vertices of $\mathcal{L}$ have a demand of n_L.
- **Costs:** all edges have a cost of zero except the edges of $E_{C \to S^h}$ where the cost of the edge between the point and the center is set according to the distance and the clustering objective (k-median or k-means). As noted earlier a vertex j will only be connected to the same color vertex that represents center i in the network flow graph, we refer to that vertex by $i^{\chi(j)}$ and clearly $i^{\chi(j)} \in S^{\chi(j)}$. Specifically, $\forall (j, i^{\chi(j)}) \in E_{C \to S^h}, \text{cost}(j, i^{\chi(j)}) = d^p(j, i)$ where $p = 1$ for the k-median and $p = 2$ for the k-means.

We write the cost for a constructed flow graph as $\sum_{j \in C, i \in S} d^p(j, i) x_{ij}$ where x_{ij} is the amount of flow between vertex j and center $i^{\chi(j)}$. Since all capacities, demands, and costs are set to integer values. Therefore we can obtain an optimal solution (maximum flow at a minimum cost) in polynomial time where all flow values are integers. Therefore, we can solve LCAL optimally for a given distribution of points.

[3] Fair allocation [8, 9, 19] is a sub-problem solved in fair clustering to finally yield a full algorithm for fair clustering.

The above construction are for the k-median and k-means. For the k-center we slightly modify the graph. First, we point out that unlike the k-median and k-means, for the k-center the objective value has only a polynomial set of possibilities (kn many exactly) since it is the distance between a center and a vertex. So our network flow diagram is identical but instead of setting a cost value for the edges in edges of $E_{C\to S^h}$, we instead pick a value d from the set of possible distances $d(j,i)$ where $j \in C, i \in S$ and draw an edge between a point j and a center $i^{\chi(j)}$ only if $d(j,i) \le d$. Also we do not need to solve the minimum cost max flow problem, instead the max flow problem is sufficient. □

4.2 Efficient Algorithms for LCAL for the Two Label Case:

For the k-median and k-means and the two label case we present an algorithm with $O(n(\log(n)+k))$ running-time. The intuition behind our algorithm is best understood for the case with "exact population proportions" for both the positive and negative labels[4]. First, we note that each color $h \in \mathcal{H}$ exists in proportion $r_h = \frac{|C^h|}{|C|}$ where we refer to r_h as the population proportion. The case of exact population proportions for the positive and negative labels, is the one where $\forall\, h \in \mathcal{H}, \forall L \in \{P,N\} : l_h^L = u_h^L = r_h = \frac{|C^h|}{|C|}$

That is, the upper and lower proportion bounds coincide and are equal to the proportion of the color in the entire set. This forces only a limited set of possibilities for the total number of points (and their colors) which we can assign to either P or N. For example, if we have two colors and $r_1 = r_2 = \frac{1}{2}$, then we can only assign an equal number of red and blue points to P and likewise to N. For the case of three colors with $r_1 = \frac{1}{3}, r_2 = \frac{1}{2}, r_3 = \frac{1}{6}$, then we can only assign points of the following form across the different labels: points for the first color = $2c$, points for the second color = $3c$, points for the third color = c where c is a non-negative integer. We refer to this smallest "atomic" number of points by n_{atomic} and the number of color h of its subset by n^h_{atomic}.

Now we define some notation $P(j) = \min_{i\in P} d(j,i)$ and $N(j) = \min_{i\in N} d(j,i)$, i.e. the distance of the closest centers to j in P and N, respectively. Further, $\phi^{-1}(P)$ and $\phi^{-1}(N)$ are the set of points assigned to the positive and negative centers by the assignment ϕ, respectively. We can now define the drop of a point j as $drop(j) = N(j) - P(j)$, clearly the larger $drop(j)$ the higher the cost goes down as we move it from the negative to the positive set. We can obtain a sorted values of $drop$ for each color in $O\big(n(\log n + k)\big)$ run-time.

The algorithm is shown (algorithm block (1)). In the first step we start with all points in N, then in step 2 we move the minimum number of n^h_{atomic} for each color h to satisfy the size bounds for each label (constraint (1c)). Finally in the loop starting at step 3, we move more points to the positive label (in an "atomic" manner) if it lowers the cost and is within the size bounds.

Theorem 2. *Algorithm (1) finds the optimal solution and runs in* $O\Big(n(\log n + k)\Big)$ *time.*

Proof. First we prove that the solution is feasible. Constraint (1b) for the color proportionality holds, this can is clearly the case before the start of the loop since the centers with negative labels cover the entire set which is color proportional and the the centers with positive labels cover cover nothing which is also color proportional. In each iteration, we move an atomic number of each color from the negative to the positive label and hence both the negative and the positive set of centers satisfy color proportionality in the points they cover.

For constraint (1b) because of exact preservation of the color proportions, we can always tighten the bounds $(LB)_L$ and $(UB)_L$ for each label L such that there multiples of n_{atomic} without modification to the problem, so we assume that $(LB)_N = a\, n_{\text{atomic}}$, $(LB)_P = b\, n_{\text{atomic}}$, $(UB)_N = a'\, n_{\text{atomic}}$, $(UB)_P = b'\, n_{\text{atomic}}$ where a, a', b, b' are non-negative integers and clearly $a \le b$ and $a' \le b'$. Step 2 satisfies the lower bound on the number of points in the positive label and the upper bound for the negative set. Note that if this step fails then the problem has infeasible constraints. Further, since we have moved the minimum number of points from the negative set to the positive set, it follows that the upper bounds on the positive are also satisfied since $(LB)_P \le (UB)_P$, also the lower bound on the negative set is also satisfied since $(LB)_N \le (UB)_N$. Finally in step 5, the size bounds are always checked fair therefore both labels are balanced.

Optimally follows since we move the points with the highest $drop$ value to the positive set (these are also the points closest to the positive set). Further, in step 5 we stop moving any points to the positive if there isn't a reduction in the clustering cost. Note that since the values in $drop$ are sorted, another iteration would not reduce the cost.

Finding the closest center of each label for every point takes $O(nk)$ time. Finding and sorting the values in $drop$ clearly takes $O(n \log n)$ time. The algorithm does constant work in each iteration for at most n many iterations. Thus, the run time is $O\big(n(\log n + k)\big)$. □

With more elaborate conditional statements, the above algorithms can be generalized to give all solution values for arbitrary choices of label size bounds (constraint(1c)) with the same asymptotic run-time. Such a solution would be useful as it would enable

Algorithm 1 Exact Preservation for k-median / k-means

1: Find an assignment ϕ_0 that assigns all points to their nearest center in N, this means that $|\phi_0^{-1}(N)| = n$ and $|\phi_0^{-1}(P)| = 0$. Set $\phi^* = \phi_0$.
2: Move $q_h = r_h \max\{(LB)_P, n - (UB)_N\}$ many points of color h with the highest values in $drop$ from the negative label to the positive label
3: **for** $i = \left(\frac{n}{\sum_{h\in\mathcal{H}} q_h}\right)$ to $\frac{n}{n_{\text{atomic}}}$ **do**
4: Take n^h_{atomic} many points from each color h with the highest values in $drop$, call the new assignment ϕ'.
5: **if** $\phi'^{-1}(P)$ and $\phi'^{-1}(N)$ are within bounds **and** $cost(\phi') < cost(\phi^*)$ **then**
6: update the assignment to $\phi^* = \phi'$
7: **else**
8: break
9: **end if**
10: **end for**

[4]The general case is shown in the appendix.

the decision maker to see the complete trade-off between the label sizes and the clustering cost (quality).

5 ALGORITHMS AND THEORETICAL GUARANTEES FOR LCUL

5.1 Computational Hardness of LCUL

We start by discussing the hardness of LCUL. In contrast to LCAL, the LCUL problem it not solvable in polynomial time. In the fact, the following theorem shows that even if we were to drop one constraint for the LCUL (problem (2)) we would still have an NP-hard problem.

Theorem 3. *For the* LCUL *problem with two labels and two colors, dropping one of the constraints(2b), (2c), or (2d) still leads to an NP-hard problem.*

Having established the hardness of LCUL for different sets of constraints, we show that it is fixed-parameter tractable[5] for a constant number of labels. This immediately follows since a given choice of labels for the centers leads to an instance of LCAL which is solvable in polynomial time and there are at most m^k many possible choice labels.

Theorem 4. *The* LCUL *problem is fixed-parameter tractable for a constant number of labels.*

It is also worth wondering if the problem remains hard if we were to drop two constraints and have only one instead. Interestingly, we show that even for the case where the number of labels m is super-constant ($m = \Omega(1)$) , if we only had the color-proportionality constraint (2b) or the constraint on the number of labels (2c), then the problem is solvable in polynomial time. However, if we only had constraint (2d) for the number of centers a label has, the problem is still NP-hard.

Theorem 5. *Even if number of labels $m = \Omega(1)$, the* LCUL *problem is solvable in polynomial time under constraint (2b) alone or constraint (2c) alone. However, it is NP-hard under constraint (2d) alone.*

5.2 A Randomized Algorithm for label proportional LCUL:

Here we consider a natural special case of the LCUL problem which we call color and label proportional case (CLP) where the constraints are restricted to a specific form. In CLP each label must have color proportions "around" that of the population, i.e. color h has proportion r_h in each label $L \in \mathcal{L}$. Further, each label has a proportion $\alpha_L \in [0, 1]$ and $\sum_{L\in\mathcal{L}} \alpha_L = 1$, this proportion decides the number of points the label covers and the number of centers it has. I.e., label L covers around $\alpha_L n$ many points and has around $\alpha_L k$ many centers. Therefore, the optimization takes on the following form below where we have included the ϵ values to relax the constraints (note that for every value of ϵ, we have that $\epsilon \geq 0$):

$$\min_{\phi,\ell} \Big(\sum_{j\in C} d^p(j, \phi(j)) \Big)^{1/p} \tag{4a}$$

$$\forall L\in\mathcal{L}, \forall h\in\mathcal{H} : (r_h - \epsilon^A_{h,L}) \sum_{\substack{i\in S:\\ \ell(i)=L}} |\, C_i \,| \leq \sum_{\substack{i\in S:\\ \ell(i)=L}} |\, C_i^h \,| \leq (r_h + \epsilon'^A_{h,L}) \sum_{\substack{i\in S:\\ \ell(i)=L}} |\, C_i \,| \tag{4b}$$

$$\forall L \in \mathcal{L} : (\alpha_L - \epsilon^B_L)n \leq \sum_{i\in S:\ell(i)=L} |\, C_i \,| \leq (\alpha_L + \epsilon'^B_L)n \tag{4c}$$

$$\forall L \in \mathcal{L} : (\alpha_L - \epsilon'^C_L)k \leq |S^L| \leq (\alpha_L + \epsilon^C_L)k \tag{4d}$$

We note that even when the constraints take on this specific form the problem is still NP-hard as shown in the theorem below:

Theorem 6. *The* CLP *problem is NP-hard even for the two color and two label case.*

We show a randomized algorithm (algorithm block (2)) which always gives an optimal cost to the clustering and satisfies all constraints in expectation and further satisfies constraint (4d) deterministically with a violation of at most 1. Our algorithm is follows three steps. In step 1 we find the assignment ϕ^* by assigning each point to its nearest center, thereby guaranteeing an optimal clustering cost. In step 2, we set the center-to-label probabilistic assignments $p^i_L = \alpha_L$. Then in step 3, we apply dependent rounding, due to Gandhi et al. [21], to the probabilistic assignments to find the deterministic assignments. This leads to the following theorem:

Theorem 7. *Algorithm 2 gives an optimal clustering and satisfies constraints (4b,4c,4d) in expectation with (4d) being satisfied deterministically at a violation at most 1.*

Proof. The optimality of the clustering cost follows immediately since each point is assigned to its closest center. Now, we show that the assignment satisfies all of the constraints. We have $p^i_L = \alpha_L$ for each center i. Now we prove that constraints (2b,2c,2d) hold in expectation over the assignments P^i_L. Note that P^i_L is also an indicator random variable for center i, taking label L. Then we can show that using property **(A)** of dependent rounding (marginal probability) that:

$$\mathbb{E}[\sum_{i\in S:\ell(i)=L} |\, C_i \,|] = \mathbb{E}[\sum_{i\in S} |\, C_i \,|\, P^i_L] = \sum_{i\in S} |\, C_i \,|\, \mathbb{E}[P^i_L]$$
$$= \sum_{i\in S} |\, C_i \,|\, p^i_L = \alpha_L \sum_{i\in S} |\, C_i \,| = \alpha_L n$$

Clearly, constraint (4c) is satisfied. Through a similar argument we can show that the rest of the constraints also hold in expectation.

We have that $\forall L \in \mathcal{L} : |S^L| = \sum_{i\in S} P^i_L = \sum_{i\in S} \alpha_L = \alpha_L k$. By property **(B)** of dependent rounding (degree preservation) we have $\forall L \in \mathcal{L} : |S^L| \in \{\lfloor \alpha_L k \rfloor, \lceil \alpha_L k \rceil\}$. Therefore constraint (4d) is satisfied in every run of the algorithm at a violation of at most 1. □

Algorithm 2 Randomized LCUL Algorithm

1: Find the assignment ϕ^* by assigning each point to its nearest center in S.
2: For each center i, set its probabilistic assignment for label L to $p^i_L = \alpha_L$.
3: Apply dependent rounding [21] to probabilistic assignments p^i_L to get the deterministic assignments P^i_L

We note that dependent rounding enjoys the **Marginal Probability** property which means that $\Pr[P^i_L = 1] = p^i_L$. This enables us to satisfy the constraints in expectation. While we note

[5] An algorithm is called fixed-parameter tractable if its run-time is $O(f(k)n^c)$ where $f(k)$ can be exponential in k, see [13] for more details.

that letting each center i take label L with probability α_L would also satisfy the constraints in expectation. Dependent rounding also has the **Degree Preservation** property which implies that $\forall L \in \mathcal{L} : \sum_{i \in S} P_L^i \in \{\lfloor \sum_{i \in S} p_L^i \rfloor, \lceil \sum_{i \in S} p_L^i \rceil\}$ which leads us to satisfy constraint (4d) deterministically (in every run of the algorithm) with a violation of at most 1. Further, dependent rounding has the **Negative Correlation** property which under some conditions leads to a concentration around the expected value. Although, we cannot theoretically guarantee that we have a concentration around the expected value, we observe empirically (section 6.2) that dependent rounding is much better concentrated around the expected value, especially for constraint (4c) for the number of points in each label.

6 EXPERIMENTS

We run our algorithms using commodity hardware with our code written in Python 3.6 using the `NumPy` library and functions from the `Scikit-learn` library [31]. We evaluate the performance of our algorithms over a collection of datasets from the UCI repository [16]. For all datasets, we choose specific attributes for group membership and use numeric attributes as coordinates with the Euclidean distance measure. Through all experiments for a color $h \in \mathcal{H}$ with population proportion $r_h = \frac{|C^h|}{|C|}$ we set the the upper and lower proportion bounds to $l_h = (1-\delta)r_h$ and $u_h = (1+\delta)r_h$, respectively. Note that the upper and lower proportion bounds are the same for both labels. Further, we have $\delta \in [0, 1]$, and smaller values correspond to more stringent constraints. In our experiments, we set δ to 0.1. For both the LCAL and LCUL we measure the price of fairness PoF $= \frac{\text{fair solution cost}}{\text{color-blind solution cost}}$ where fair solution cost is the cost of the fair variant and color-blind solution cost is the cost of the "unfair" algorithm which would assign each point to its closest center.

We note that since all constraints are proportionality constraints, we calculate the proportional violation. To be precise, for the color proportionality constraint (2b), we consider a label L and define $\Delta_h^L \in [0, 1]$ where Δ_h^L is the smallest relaxation of the constraint for which the constraint is satisfied, i.e. the minimum value for which the following constraint is feasible given the solution: $(l_h^L - \Delta_h^L) \sum_{i \in S:\ell(i)=L} |C_i| \leq \sum_{i \in S:\ell(i)=L} |C_i^h| \leq (u_h^L + \Delta_h^L) \sum_{i \in S:\ell(i)=L} |C_i|$, having found Δ_h^L we report Δ_{color} where $\Delta_{\text{color}} = \max_{\{h \in \mathcal{H}, l \in \mathcal{L}\}} \Delta_h^L$. Similarly, we define the proportional violation for the number of points $\Delta^L_{\text{points/label}}$ assigned to a label as the minimal relaxation of the constraint for it to be satisfied. We set $\Delta_{\text{points/label}}$ to the maximum across the two labels. In a similar manner, we define $\Delta_{\text{center/label}}$ for the number of centers a label receives.

We use the k-means++ algorithm [5] to open a set of k centers. These centers are inspected and assigned a label. Further, this set of centers and its assigned labels are fixed when comparing to baselines other than our algorithm.

Clustering Baseline: In the labeled setting and in the absence of our algorithm, the only alternative that would result in. a fair outcome is a fair clustering algorithm. Therefore we compare against fair clustering algorithms. The literature in fair clustering is vast, we choose the work of [8] as it can be tailored easily to this setting in which the centers are open. Further, it allows both lower and upper proportion bounds in arbitrary metric spaces and results in fair solutions at relatively small values of PoF compared to larger PoF (as high as 7) reported in [12]. Our primary concern here is not to compare to all fair clustering work, but gauge the performance of these algorithms in this setting. We also compare against the "unfair" solution that would simply assign each point to its closest center which we call the nearest center baseline. Though this in general would violate the fairness constraints it would result in the minimum cost.

Datasets: We use two datasets from the UCI repository: The **Adult** dataset consisting of 32,561 points and the **CreditCard** dataset consisting of 30,000 points. For the group membership attribute we use race for **Adult** which takes on 5 possible values (5 colors) and marriage for **CreditCard** which takes on 4 possible values (4 colors). For the **Adult** dataset we use the numeric entries of the dataset (age, final-weight, education, capital gain, and hours worked per week) as coordinates in the space. Whereas for the **CreditCard** dataset we use age and 12 other financial entries as coordinates.

6.1 LCAL Experiments

Adult Dataset: After obtaining k centers using the k-means++ algorithm, we inspect the resulting centers. In an advertising setting, it is reasonable to think that advertisements for expensive items could be targeting individuals who obtained a high capital gain. Therefore, we choose centers high in the capital gain coordinate to be positive (assign an advertisement for an expensive item). Specifically, centers whose capital gain coordinate is $\geq$ 1,100 receive a positive label and the remaining centers are assigned a negative one. Such a choice is somewhat arbitrary, but suffices to demonstrate the effectiveness of our algorithm. In real world scenarios, we expect the process to be significantly more elaborate with more representative features available. We run our algorithm for LCAL as well as the fair clustering algorithm as a baseline. Figure 1 shows the results. It is clear that our algorithm leads to a much smaller PoF and the PoF is more robust to variations in the number of clusters. In fact, our algorithm can lead to a PoF as small as 1.0059 (0.59%) and very close to the unfair nearest center baseline whereas fair clustering would have a PoF as large as 1.7 (70%). Further, we also see that the unlike the nearest center baseline, fair labeled clustering has no proportional violations just like fair clustering.

Here for the LCAL setting, we compare to the optimal (fairness-agnostic) solution where each point is simply routed to its closest center regardless of color or label. We use the same setting at that from section 6. We set $\delta = 0.1$ and measure the PoF. Since the (fairness-agnostic) solution does not consider the fairness constraint we also measure its proportional violations. Figures 6 and 7 show the results over the **Adult** and **CreditCard** datasets. We can clearly see that although the (fairness-agnostic) solution has the smallest cost it has large color violation. We also see that our algorithm unlike fair clustering achieves fairness but at a much lower PoF.

CreditCard Dataset: Similar to the **Adult** dataset experiment, after finding the centers using k-means++, we assign them positive and negative labels. For similar motivations, if the center has a coordinate corresponding to the amount of balance that is $\geq$ 300,000

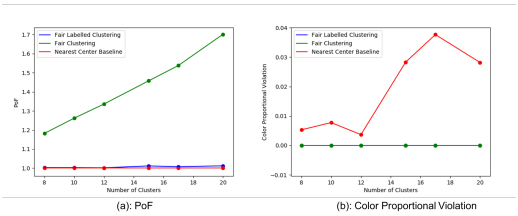

Figure 1: Adult dataset results (a): PoF, **(b):**Δ_{color}

we assign the center a positive label and a negative one otherwise. Figure 2 shows the results of the experiments. We see again that our algorithm leads to a lower price of fairness than fair clustering, but not to the same extent as in the **Adult** dataset but it still has no proportional violation just like fair clustering.

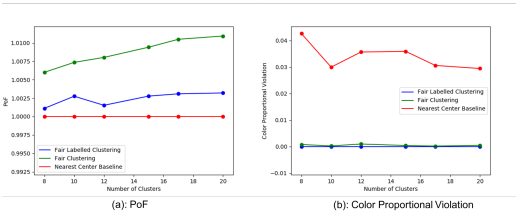

Figure 2: CreditCard dataset results (a): PoF, **(b):**Δ_{color}

As mentioned in section 4.2, algorithm (1) can allow the user to obtain the solutions for different values of $|\phi^{-1}(P)|$ (the number of points assigned to the positive set) without an asymptotic increase in the running time. In figure 3 we show a plot of $|\phi^{-1}(P)|$ vs the clustering cost. Interestingly, requiring more points to be assigned to the positive label comes at the expense of a larger cost for some instances (**Adult** with $k = 15$) whereas for others it has a non-monotonic behaviour (**Adult** with $k = 10$). This can perhaps be explained by the different choices of centers as k varies. There are 5 centers with positive labels for $k = 10$ (50% of the total), but only 4 for $k = 15$ (less than 30%) making it difficult to route points to positive centers.

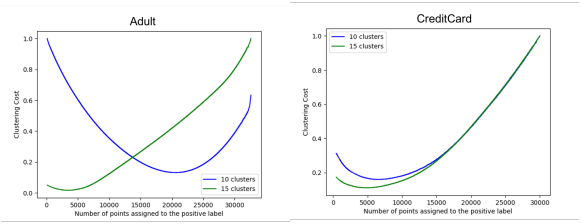

Figure 3: A plot of $|\phi^{-1}(P)|$ vs the clustering cost (normalized by the maximum cost obtained).

6.2 LCUL Experiments

Similar to the LCAL setting for LCUL we get the centers by running k-means++. However, we do not have the labels. We compare our algorithm (algorithm 2) to two baselines: (1) Nearest Center with Random Assignment (**NCRA**) and (2) Fair Clustering (**FC**). We refer to our algorithm (block 2) as **LFC** (labeled fair clustering). In **NCRA** we assign each point to its closest center which leads to an optimal clustering cost, whereas for fair clustering (**FC**) we solve the fair clustering problem. For both **NCRA** and **FC** we assign each center label L with probability α_L.

We use two labels with $\alpha_1 = \frac{1}{4}$ and $\alpha_2 = \frac{3}{4}$. For all colors and labels we set $\epsilon^A_{h,L} = \epsilon'^A_{h,L} = 0.2$ and for all labels we set $\epsilon^B_L = \epsilon'^B_L = \epsilon^C_L = \epsilon'^C_L = 0.1$. Further, all algorithms satisfy the constraints in expectation, therefore we seek a measure of centrality around the expectation like the variance. Each algorithm is ran 50 times and we report the average values of Δ_{color},$\Delta_{\text{points/label}}$, and $\Delta_{\text{center/label}}$.

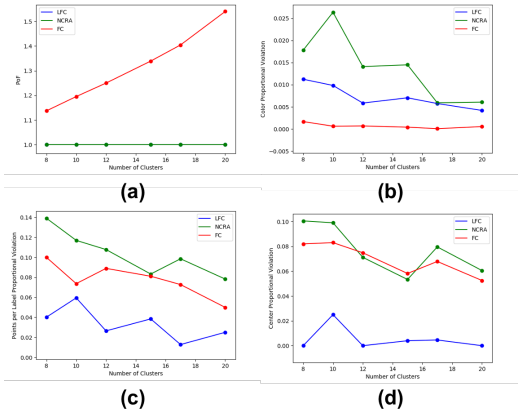

Figure 4: LCUL **results on the Adult dataset. (a):**PoF, **(b):**Δ_{color}, **(c):**$\Delta_{\text{points/label}}$ **,(d):**$\Delta_{\text{center/label}}$.

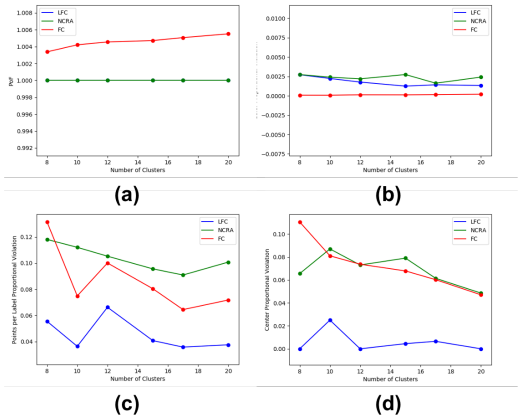

Figure 5: LCUL **results on the CreditCard dataset. (a):**PoF, **(b):**Δ_{color}, **(c):**$\Delta_{\text{points/label}}$ **,(d):**$\Delta_{\text{center/label}}$.

Figures 4 and 5 show the results for **Adult** and **CreditCard**. For PoF, our algorithm achieves an optimal clustering and hence coincides with **NCRA** whereas fair clustering achieves a much higher PoF as large as 1.5. For the color proportionality (Δ_{color}),

we see that fair clustering has almost no violation whereas the **NCRA** and labeled clustering have small but noticeable violations. For the number of points a label receives ($\Delta_{\text{points/label}}$) we notice that all algorithms have a violation although labeled clustering has a smaller violation mostly. As noted earlier, we suspect that this is a result of dependent rounding's negative correlation property leading to some concentration around the expectation. Finally, for the number of centers a label receives ($\Delta_{\text{center/label}}$), clearly **LFC** has a much lower violation.

6.3 Algorithm Scalability

Here we investigate the scalability of our algorithms. In particular, we take the **Census1990** dataset which consists of 2,458,285 points and sub-sample it to a specific number, each time we find the centers with the k-means algorithm[6], assign them random labels, and solve the LCAL and LCUL problems. Note since we care only about the run-time a random assignment of labels should suffice. Our group membership attribute is gender which has two values (two colors). We find our algorithm are indeed highly scalable (figure 6) and that even for 500,000 points it takes less than 90 seconds. We note in contrast that the fair clustering algorithm of [8] would takes around 30 minutes to solve a similar size on the same dataset. In fact, scalability is an issue in fair clustering and it has instigated a collection of work such as [6, 26]. The fact that our algorithm performs relatively well run-time wise is worthy of noting.

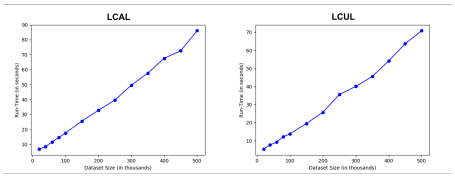

Figure 6: Dataset size vs algorithm Run-Time: (left) LCAL**, (right)** LCUL**.**

7 CONCLUSION

Motivated by fairness considerations and the quality of outcome each cluster receives, we have introduced fair labeled clustering. We showed algorithms for the case where the centers' labels are decided and have shown that unlike fair clustering we end up with a much lower cost while still satisfying the fairness constraints. For the case where the centers' labels are not decided we gave a detailed characterization of the complexity and showed an algorithm for a special case. Experiments have shown that our algorithms are scalable and much faster than fair clustering.

8 ACKNOWLEDGMENTS

This research was supported in part by NSF CAREER Award IIS-1846237, NSF Award CCF-1749864, NSF Award CCF-1852352, NSF Award SMA-2039862, NIST MSE Award #20126334, DARPA GARD #HR00112020007, DoD WHS Award #HQ003420F0035, DARPA SI3-CMD #S4761, ARPA-E DIFFERENTIATE Award #1257037, and gifts by research awards from Adobe, Amazon, and Google.

[6]We choose $k = 5$ for all different dataset sizes.

REFERENCES

[1] Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. 2020. Fair clustering via equitable group representations. arXiv:2006.11009 [cs.LG]
[2] Charu Chandra Aggarwal, Joel Leonard Wolf, and Philip Shi-lung Yu. 2004. Method for targeted advertising on the web based on accumulated self-learning data, clustering users and semantic node graph techniques. US Patent 6,714,975.
[3] Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. 2019. Clustering without over-representation. In *International Conference on Knowledge Discovery and Data Mining*.
[4] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine bias. ProPublica. *See https://www. propublica. org/article/machine-bias-risk-assessments-in-criminal-sentencing* (2016).
[5] D Arthur and S Vassilvitskii. 2007. k-means++: The Advantages of Careful Seeding. *ACM-SIAM Symposium on Discrete Algorithms*.
[6] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. 2019. Scalable fair clustering. *International Conference on Machine Learning*.
[7] Xiaohui Bei, Shengxin Liu, Chung Keung Poon, and Hongao Wang. 2020. Candidate Selections with Proportional Fairness Constraints. In *International Conference On Autonomous Agents and Multi-Agent Systems*.
[8] Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. 2019. Fair algorithms for clustering. In *Neural Information Processing Systems*.
[9] Ioana O Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R Schmidt, and Melanie Schmidt. 2019. On the cost of essentially fair clusterings. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*.
[10] Brian Brubach, Darshan Chakrabarti, John P Dickerson, Samir Khuller, Aravind Srinivasan, and Leonidas Tsepenekas. 2020. A Pairwise Fair and Community-preserving Approach to k-Center Clustering. *International Conference on Machine Learning*.
[11] Daqing Chen, Sai Laing Sain, and Kun Guo. 2012. Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining. *Journal of Database Marketing & Customer Strategy Management*.
[12] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2017. Fair clustering through fairlets. In *Neural Information Processing Systems*.
[13] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. 2015. *Parameterized algorithms*. Vol. 5. Springer.
[14] Amit Datta, Anupam Datta, Jael Makagon, Deirdre K Mulligan, and Michael Carl Tschantz. 2018. Discrimination in online advertising: A multidisciplinary inquiry. In *ACM Conference on Fairness, Accountability, and Transparency*.
[15] Ian Davidson and SS Ravi. 2020. Making existing clusterings fairer: Algorithms, complexity results and insights. In *AAAI Conference on Artificial Intelligence*.
[16] Dheeru Dua and Casey Graff. 2017. UCI machine learning repository. (2017).
[17] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Innovations in Theoretical Computer Science Conference*.
[18] Seyed A Esmaeili, Brian Brubach, Aravind Srinivasan, and John P Dickerson. 2021. Fair Clustering Under a Bounded Cost. *arXiv preprint arXiv:2106.07239* (2021).
[19] Seyed A Esmaeili, Brian Brubach, Leonidas Tsepenekas, and John P Dickerson. 2020. Probabilistic Fair Clustering. *Neural Information Processing Systems*.
[20] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In *International Conference on Knowledge Discovery and Data Mining*.
[21] Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. 2006. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)* 53, 3 (2006), 324–360.
[22] Michael R Garey and David S Johnson. 1979. *Computers and intractability*. Vol. 174. freeman San Francisco.
[23] Mehrdad Ghadiri, Samira Samadi, and Santosh Vempala. 2021. Socially fair k-means clustering. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 438–448.
[24] Jiawei Han, Micheline Kamber, and Jian Pei. 2011. Data mining concepts and techniques third edition. *The Morgan Kaufmann Series in Data Management Systems* 5, 4 (2011), 83–124.
[25] David Harris, Shi Li, Aravind Srinivasan, Khoa Trinh, and Thomas Pensyl. 2018. Approximation algorithms for stochastic clustering. In *Neural Information Processing Systems*.
[26] Lingxiao Huang, Shaofeng Jiang, and Nisheeth Vishnoi. 2019. Coresets for clustering with fairness constraints. In *Neural Information Processing Systems*.
[27] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. 2019. Fair k-center clustering for data summarization. *International Conference on Machine Learning*.
[28] Ava Kofman and Ariana Tobin. 2019. Facebook Ads Can Still Discriminate Against Women and Older Workers, Despite a Civil Rights Settlement. (2019). https://www.propublica.org/article/facebook-ads-can-still-discriminate-

against-women-and-older-workers-despite-a-civil-rights-settlement
[29] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*.
[30] Deepak P. 2020. Whither Fair Clustering? *arXiv preprint arXiv:2007.07838*.
[31] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine Learning research*.
[32] David B Shmoys, Chaitanya Swamy, and Retsef Levi. 2004. Facility location with service installation costs. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*. 1088–1097.
[33] Till Speicher, Muhammad Ali, Giridhari Venkatadri, Filipe Nunes Ribeiro, George Arvanitakis, Fabrício Benevenuto, Krishna P Gummadi, Patrick Loiseau, and Alan Mislove. 2018. Potential for discrimination in online targeted advertising. In *ACM Conference on Fairness, Accountability, and Transparency*.
[34] Pang-Ning Tan, Michael Steinbach, DA Karpatne, and DV Kumar. 2018. Introduction to Data Mining , 2nd Editio.
[35] United States Senate. 1991. S. 1745 – 102nd Congress: Civil Rights Act of 199. https://www.govtrack.us/congress/bills/102/s1745.
[36] Dachuan Xu and Shuzhong Zhang. 2008. Approximation algorithm for facility location with service installation costs. *Operations Research Letters* 36, 1 (2008), 46–50.
[37] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *International Conference on Machine Learning*.

A OMITTED PROOFS

We note that all of our hardness results use the k-center problem for simplicity. Before we introduce the hardness result, we note all of our reductions are from exact cover by 3-sets (X3C) [22] where we have universe $\mathcal{U} = \{u_1, u_2, \ldots, u_{3q}\}$ and subsets $\mathcal{W}_1, \ldots, \mathcal{W}_t$ where $t = q + r$ and for non-trivial instances $r > 0$. We form an instance of LCUL by representing each one the subsets $\mathcal{W}_1, \ldots, \mathcal{W}_t$ by a vertex and each element in $\mathcal{U} = \{u_1, u_2, \ldots, u_{3q}\}$ by a vertex. The centers are the sets $\mathcal{W}_1, \ldots, \mathcal{W}_t$ and they are given a blue color whereas the rest of the points (in $\mathcal{U}$) are red. Further, each point u_i is connected by a edge to a center $\mathcal{W}_i$ if and only if $u_i \in \mathcal{W}_j$. The distances between any two points is the length of the shortest path between them. This clearly leads to a metric. See figure 7 for an example. This is essentially a reduction we follow in all proofs, sometimes changes are introduced and mentioned explicitly in the proofs.

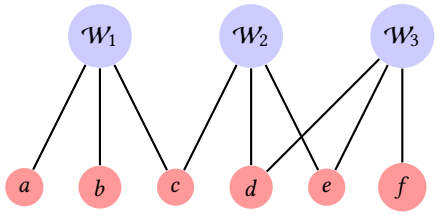

Figure 7: Example of the reduction for theorem (3). This is an instance of the LUCL problem for an instance $U = \{a, b, c, d, e, f\}$, $\mathcal{W}_1 = \{a, b, c\}$, $\mathcal{W}_2 = \{c, d, e\}$ and $\mathcal{W}_3 = \{d, e, f\}$ with $q = 2$, $|U| = 3q$ and $t = 3$.

Now we introduce the following theorem:

Theorem 8. *Even if the color-proportionality constraint (2b) are ignored*[7] LCUL *is NP-hard.*

Proof. As mentioned we consider an instance of exact cover by 3-sets (X3C) with universe $\mathcal{U} = \{u_1, u_2, \ldots, u_{3q}\}$ and subsets $\mathcal{W}_1, \ldots, \mathcal{W}_t$. We construct an instance of LCUL where the proportionality constraints are ignored. Further, we only have two labels $\mathcal{L} = \{N, P\}$, we set $(CL)_P = 0, (CU)_P = q, (CL)_N = 0, (CU)_N = t$ and $(LB)_P = 4q$, $(UB)_P = 3q + t$, $(LB)_N = 0$, $(UB)_N = 3q + t$.

A solution for X3C leads to a solution for LCUL at cost 1: Take the collection of q many subsets that solve X3C and give their corresponding centers in LCAL a positive label. Then it is clear that $|S^P| = q$ and that the number of points covered by the positive centers is $4q$ and that this done at a cost of 1. The centers that do not correspond to the solution of X3C will be given a negative label and assigned no points.

A solution for LCUL at cost 1 leads to a solution X3C: A solution for LCUL cannot assign more than $(CU)_P = q$ many centers a positive label and it has to cover $3q$ more points to have a total of $4q$ points and this has to be done at a distance of 1. By construction, since each center is connected to 3 points, the LCUL solution cannot have less than q centers. Further, to have $4q$ points, then each center would have to cover a unique set of 3 points at a distance of 1. Since points are connected to centers at a distance of 1 only if they are corresponding values are contained in the subsets corresponding to those centers, it follows that the q subsets in the LCUL solution are indeed an exact cover for X3C.

□

Here we instead we ignore the constraints on the number of points a label should receive, i.e. constraints (2c and keep the proportionality constraints. We show that this also results in an NP-hard problem as demonstrated in the theorem below:

Theorem 9. *Even if we do not specify the number of points a label should receive (constraint(2c)),* LCUL *is NP-hard.*

Proof. Similar to the proof of theorem (8) we follow the reduction from X3C with two labels for LCUL, i.e. $\mathcal{L} = \{N, P\}$, but now we consider the color of the vertices. Vertices of the subsets $\mathcal{W}_1, \ldots, \mathcal{W}_t$ are blue and all of the vertices of the elements of $\mathcal{U}$ are red. For the LCUL instance, we set $(CL)_P = q$, $(CU)_P = t$,$(CL)_N = 0$, $(CU)_N = t$. The representation for the negative set is ignored, i.e. $l^N_{\text{red}} = l^N_{\text{blue}} = 0$ and $u^N_{\text{red}} = u^N_{\text{blue}} = 1$. For the positive set, we only have set a bound on the lower proportion for the red color, specifically $l^P_{\text{red}} = \frac{3}{4}$, $u^P_{\text{red}} = 1$ and $l^P_{\text{blue}} = 0, u^P_{\text{blue}} = 1$. As the reduction of theorem (8) the optimal value of the k-center objective cannot be less than 1.

A solution for X3C leads to a solution for LCUL at cost 1: Take the q subsets in the solution of X3C and assign their corresponding centers a positive labels, then $|S^P| = q \geq (CU)_P$. Further since elements of $\mathcal{U}$ are represented by red vertices, you will have $3q$ red vertices covered at a distance of 1, the red proportion of the positive label would be $\frac{3q}{4q} = \frac{3}{4} \geq l^P_{\text{red}}$. To complete the solution assign the rest of the centers a negative label.

A solution for LCUL at cost 1 leads to a solution X3C: A solution for LCUL would have to choose at least $(CL)_P = q$ many centers. Since all centers are blue and because there are only $3q$ many red points in the graph, we would have to choose exactly q centers and cover all of the $3q$ many red points to satisfy the color proportionality constraints of l^P_{red}. Since this is being done at a cost of 1, these points must be representing elements in $\mathcal{U}$ that are contained in the subsets corresponding to the selected centers. Further, since every center is connected to exactly 3 points at radius 1, we have found an exact cover. □

Theorem 10. *Even if we do not specify the number of centers of each label (ignoring constraints (2d)),* LCUL *is NP-hard.*

Proof. Similar to theorems (8,9) we follow the same reduction from X3C. This time we ignore constraint (2d) on the number of centers, i.e. $0 \leq |S^N|, |S^P| \leq k$. We set $(LB)_P = (UB)_P = 4q$ and $(LB)_N = 0, (LB)_N = n$. Further for the color proportionality constraints, we have for the positive set we set $l^P_{\text{red}} = u^P_{\text{red}} = \frac{3}{4}$, $l^P_{\text{blue}} = u^P_{\text{blue}} = \frac{1}{4}$ and for the negative set we have $l^N_{\text{red}} = l^N_{\text{blue}} = 0$. $u^N_{\text{red}} = u^N_{\text{blue}} = 1$.

A solution for X3C leads to a solution for LCUL at cost 1: Simply let the subsets (centers) in the solution if X3C have a positive label and assign all of the points in $\mathcal{U}$ to them. Clearly, we have $(LB)_P = (UB)_P = 4q$ and the red color has a representation of $\frac{3}{4}$ and the blue has a representation of $\frac{1}{4}$. Furthe, this is done at an optimal cost of 1.

[7] We can simply remove the constraint or set $l^L_h = 0, u^L_h = 1, \forall\, h \in \mathcal{H}, L \in \mathcal{L}$.

A solution for LCUL at cost 1 leads to a solution X3C: Since $(LB)_P = (UB)_P = 4q$, $l^P_{\text{red}} = u^P_{\text{red}} = \frac{3}{4}$, and $l^P_{\text{blue}} = u^P_{\text{blue}} = \frac{1}{4}$, it follows that the positive set should cover $\frac{3}{4}4q = 3q$ many red points and that it must also cover $\frac{1}{4}4q = q$ many blue points. Since all blue points are centers and all red points are from $\mathcal{U}$,it follows that we have to choose q many centers to cover $3q$ many points at an optimal cost of 1. This leads to a solution for X3C. □

Now we re-state the original theorem from the main paper:

Theorem 3. *For the* LCUL *problem with two labels and two colors, dropping one of the constraints(2b), (2c), or (2d) still leads to an NP-hard problem.*

Proof. This follows immediately from theorems (8,9,10) above. □

Theorem 4. *The* LCUL *problem is fixed-parameter tractable for a constant number of labels.*

Proof. This follows simply by noting that if the labels are assigned, then we have an LCAL instance which solvable in time that is polynomial in n and k, since $k \leq n$, it follows that the run time for solving LCAL is $O(n^c)$ for some constant c. Now, since there are at most m^k many label choices for the centers, it follows that the run time is for LCUL is $O(m^k n^c)$. □

Theorem 5. *Even if number of labels $m = \Omega(1)$, the* LCUL *problem is solvable in polynomial time under constraint (2b) alone or constraint (2c) alone. However, it is NP-hard under constraint (2d) alone.*

Proof. Let us consider the color proportionality constraint (2b) alone. To solve the problem optimally and satisfy the constraint, simply assign all points to their closest center and let all centers take one label from the set $\mathcal{L}$.

Now, we consider only the constraints on the number of centers for each label (2d). Again we assign each point to its closest center for an optimal cost. To satisfy constraints (2d), assuming the constraint parameters of (2d) lead to a feasible problem, then each label $L \in \mathcal{L}$, assign it $(CL)_L$ many centers arbitrarily. If some centers have not been assigned any labels, then simply go to label L which has not reached its upper bound $(CU)_L$ and assign more labels from it. We simply keep assigning labels from label values that have not reached their upper bound on the number of centers until all centers have a label.

Now, we consider only the constraints on the number of points a label receives (2c). We simply follow the same reduction from theorems (8,9,10), see also the beginning of this subsection for the details of the reduction from X3C. We have $t = q + r$ many subsets, we let the number of labels of the LCUL instance be $m = t = q + r$. Further, we partition the set of labels into two, i.e. $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$ where $|\mathcal{L}_1| = q$ and $|\mathcal{L}_2| = r$, and we set the lower and upper bounds for the labels according to these sets. Specifically, $\forall L \in \mathcal{L}_1 : (LB)_L = (UB)_L = 4q$ and $\forall L \in \mathcal{L}_2 : (LB)_L = (UB)_L = 1$. Now, clearly a solution for X3C leads to a solution for the LCUL instance, we simply let the subsets (centers) in the solution of X3C be the centers for the label set $\mathcal{L}_1$. Each center is assigned a label from $\mathcal{L}_1$ and covers itself and 3 points from $\mathcal{U}$, this leads to $4q$ many points which clearly satisfies the upper and lower bounds. Further, the centers not the solution are assigned a label from $\mathcal{L}_2$ and cover themselves, which is just 1 point and therefore satisfies the constraints. Now for the reverse direction, consider the set $\mathcal{L}_2$ where we have r many labels each covering 1 point. It clear, the smallest cost would be for a center to be assigned to itself, it follows that we are looking for r many centers and that each center should only be assigned to itself. This then leaves us with q many centers, since no center can cover more than $4q$ many points at a distance of 1, and since we have q many labels with each having to cover $4q$ many points, we clearly have a set cover, i.e. a solution for X3C. □

Theorem 6. *The* CLP *problem is NP-hard even for the two color and two label case.*

Proof. We follow a reduction for X3C (see the beginning of the appendix). We consider the two label case, $\mathcal{L} = \{N, P\}$. Similiar to the previous reductions we will have t many blue centers for the subsets $\mathcal{W}_1, \ldots, \mathcal{W}_t$ each being connected to its elements in $\mathcal{U}$ at a distance of 1 with all elements in $\mathcal{U}$ being red. Note that $|\mathcal{U}| = q$ and that $t = q + r$. Now we also add $2q$ many blue centers which are not connected to anything by an edge, expect for one center which is connected by an edge to a new $3(r + 2q)$ many red points, this means that any one of these red points is at a distance of 1 from this new center. Note that the increase in the problem size is still polynomial in the original X3C problem. We set the color proportionality constraint so that each label should have exactly 3:1 ratio of red points to blue points. Now the total number of points in the problem is $n = 4q + r + 2q + 3(r + 2q) = 4(3q + r)$. The number of centers $k = q + r + 2q = 3q + r$. Further, we set $\alpha_P = \frac{q}{(3q+r)}$ and $\alpha_N = 1 - \alpha_P = \frac{2q+r}{3q+r}$. We set the lower and upper size bounds according to α_P and α_N, this leads to $(LB)_P = (UB)_P = \alpha_P n = \frac{q}{(3q+r)}n = \frac{q}{(3q+r)}4(3q + r) = 4q$ and $(LB)_N = (UB)_N = \frac{2q+r}{3q+r}n = \frac{2q+r}{3q+r}4(3q + r) = 4(2q + r)$. Further, the number of centers for each label are $(CL)_P = (CU)_P = \alpha_P k = \frac{q}{(3q+r)}k = \frac{q}{(3q+r)}3q + r = q$ and $(CL)_N = (CU)_N = \alpha_N k = \frac{2q+r}{3q+r}3q + r = 2q + r$.

A solution for X3C leads to a solution for LCUL at cost 1: Simply let the q many centers representing the solution set in $\mathcal{W}_1, \ldots, \mathcal{W}_t$ be the positive labeled centers and assign them the points that belong to them and let all other centers be negative and assign the last new center all of the $3(r + 2q)$ many red children points. We then q many positive centers covering $4q$ many points with the color proportionality being 3:1 red points to blue points. Similarly, for the negative set we have $2q + r$ many centers covering $4(2q + r)$ many points at a color proportionality of 3:1 red to blue. This is done at cost of 1, so clearly optimal.

A solution for LCUL at cost 1 leads to a solution X3C: Suppe the new blue center with $3(r + 2q)$ many red children is assigned a positive label, this to achieve an optimal cost all of its children have to be assigned to it. This means that the positive set would have at least $3(r + 2q) = 6q + 3r$ many points, but $(LB)_P = (UB)_P = \alpha_P n = 4q < 6q < 6q + 3r$ which causes a contradiction. Therefore that center can never be positive. Therefore, we are looking for $\alpha_P k = q$ many centers to cover $\alpha_P n = 4q$ many points and because of the color proportionality constraint $3q$ many of them are red and q are blue. Finding this set at an optimal cost is a solution for X3C. □