

# Fair Representation Clustering with Several Protected Classes

Zhen Dai  
University of Chicago  
Chicago, IL, United States  
zhen9@uchicago.edu

Yury Makarychev  
TTIC  
Chicago, IL, United States  
yury@ttic.edu

Ali Vakilian  
TTIC  
Chicago, IL, United States  
vakilian@ttic.edu

## ABSTRACT

We study the problem of fair  $k$ -median where each cluster is required to have a fair representation of individuals from different groups. In the fair representation  $k$ -median problem, we are given a set of points  $X$  in a metric space. Each point  $x \in X$  belongs to one of  $\ell$  groups. Further, we are given fair representation parameters  $\alpha_j$  and  $\beta_j$  for each group  $j \in [\ell]$ . We say that a  $k$ -clustering  $C_1, \dots, C_k$  fairly represents all groups if the number of points from group  $j$  in cluster  $C_i$  is between  $\alpha_j|C_i|$  and  $\beta_j|C_i|$  for every  $j \in [\ell]$  and  $i \in [k]$ . The goal is to find a set  $C$  of  $k$  centers and an assignment  $\phi : X \rightarrow C$  such that the clustering defined by  $(C, \phi)$  fairly represents all groups and minimizes the  $\ell_1$ -objective  $\sum_{x \in X} d(x, \phi(x))$ .

We present an  $O(\log k)$ -approximation algorithm that runs in time  $n^{O(\ell)}$ . Note that the known algorithms for the problem either (i) violate the fairness constraints by an additive term or (ii) run in time that is exponential in both  $k$  and  $\ell$ . We also consider an important special case of the problem where  $\alpha_j = \beta_j = \frac{f_j}{f}$  and  $f_j, f \in \mathbb{N}$  for all  $j \in [\ell]$ . For this special case, we present an  $O(\log k)$ -approximation algorithm that runs in  $(kf)^{O(\ell)} \log n + \text{poly}(n)$  time.

## CCS CONCEPTS

• **Social and professional topics;** • **Mathematics of computing;**

## KEYWORDS

fair  $k$ -median, clustering, approximation algorithm, randomized algorithm

## ACM Reference Format:

Zhen Dai, Yury Makarychev, and Ali Vakilian. 2022. Fair Representation Clustering with Several Protected Classes. In *2022 ACM Conference on Fairness, Accountability, and Transparency (FAcT '22)*, June 21–24, 2022, Seoul, Republic of Korea. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3531146.3533146>

## 1 INTRODUCTION

Algorithmic decision making is widely used for high-stake decisions like college admissions [36] and criminal justice [17, 29]. While automated decision-making processes are often very efficient, there are serious concerns about their fairness. Consequently, in recent

years, there has been an extensive line of research on fairness of algorithms and machine learning approaches [18, 28, 30].

In this paper, we study the “fair representation” clustering problem proposed in the seminal work of Chierichetti et al. [15]. The notion, which is motivated by the concept of *disparate impact* [22], requires that each protected class has an approximately equal representation in each cluster. In many scenarios, a different set of benefits are associated with each cluster of points output by the algorithm. Then, it is desirable that different groups of individuals (e.g., men or women) receive the benefits associated with each of the clusters (e.g., mortgage options) in similar proportions. Further, clustering is often used for feature engineering. In this case, we need to ensure that the generated features are fair; that is, they neither introduce new nor amplify existing biases in the data set. Now, we formally define the notion of representation fairness for clustering.

*Definition 1.1 (fair representation clustering).* Given a set of points  $X$  that come from  $\ell$  different groups  $X_1, \dots, X_\ell$ , a  $k$ -clustering  $C_1, \dots, C_k$  of  $X$  is *fair* with respect to the fairness requirement specified by  $\{\alpha_j, \beta_j\}_{j \in \ell}$  if

$$\forall i \in [k], j \in [\ell], \quad \alpha_j|C_i| \leq |C_i \cap X_j| \leq \beta_j|C_i| \quad (1)$$

In fair  $k$ -median with fairness requirement  $\{\alpha_j, \beta_j\}_j$ , the goal is to find  $k$  clusters  $C_1, \dots, C_k$  and  $k$  centers,  $c_1, \dots, c_k$  (one center for each cluster) so that the clustering  $C_1, \dots, C_k$  is fair with respect to the fairness requirement and the  $\ell_1$ -objective  $\sum_{i=1}^k \sum_{x \in C_i} d(x, c_i)$  is minimized. We will say that points in  $C_i$  are assigned to center  $c_i$ . We let  $\phi$  be the assignment function that maps each point  $u$  to the center  $u$  is assigned to. To specify a solution, it is sufficient to provide the set of centers and  $\phi$ .

Bera et al. [8] and Bercea et al. [9] independently introduced this notion of fairness, which generalizes the notions studied by [2, 4, 15, 39]. Bera et al. presented a constant factor approximation algorithm for the fair representation clustering with the general  $\ell_p$ -objective. However, their algorithm returns a clustering that satisfies the fairness requirements with some additive error. When the maximum number of groups/classes to which a point may belong is  $\Delta$ , the additive error/violation is at most  $4\Delta + 3$ ; in the most common case of  $\Delta = 1$ , the additive violation is at most 3. Bercea et al. also gave constant factor approximation algorithms for a variety of clustering objective (including  $k$ -median and  $k$ -means) that violate the fairness requirement only by a small additive value. More recently, Bandyapadhyay et al. [5] designed algorithms that compute a constant factor approximation for fair  $k$ -median and  $k$ -means that run in time  $(k\Delta)^{O(k\Delta)} \text{poly}(n)$ ; these algorithms do not violate the fairness constraints. However, one of the main questions in the area of fair clustering still remains open:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

FAcT '22, June 21–24, 2022, Seoul, Republic of Korea

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9352-2/22/06...\$15.00

<https://doi.org/10.1145/3531146.3533146>

*What is the best polynomial-time approximation algorithm for representation fair clustering?*

We also study *exact* fair representation clustering, which is an important special case of the problem.

*Definition 1.2 (exact fairness).* Assume that we are given a set of points  $X$  that come from  $\ell$  disjoint groups:  $X = X_1 \cup \dots \cup X_\ell$ . A  $k$ -clustering  $C_1, \dots, C_k$  of  $X$  is *exactly fair* if

$$\forall i \in [k], j \in [\ell], \quad |C_i \cap X_j| = \frac{|X_j|}{|X|} \cdot |C_i| \quad (2)$$

We define *fairlet* as a minimal size non-empty set of points that is exactly fair. Note that all fairlets have the same size (when  $X$  is fixed). Denote this size by  $f$ . Further, for every  $j \in [\ell]$ , let  $f_j$  denote the number of points from group  $j$  in any fairlet.

This notion was previously studied for (1)  $k$ -center [9, 38] and (2)  $k$ -clustering with  $\ell_p$ -objective on balanced instances (instances with  $f_j = 1$  for all  $j$  and  $f = \ell$ ) [10]. In all these special cases of exact fairness, the fair clustering problem admits a constant factor approximation.

*Other Related Work.* Fair clustering is an active domain of research and by now it has been studied under various standard notions including both group fairness and individual fairness, e.g., [1, 8, 11, 12, 14–16, 19, 20, 23–26, 31, 32, 34, 35, 37, 41].

## 1.1 Our Results

In this paper, we study the fair representation  $k$ -median problem and give a randomized  $O(\log k)$ -approximation algorithm for it that runs in time  $n^{O(\ell)}$ . Importantly, we get a polynomial time algorithm for every fixed  $\ell$  – this is the case in most practical settings of interest for fairness applications. In fact, we design an algorithm that can handle arbitrary fairness profiles (see below for the definitions). Further, we design a much faster randomized algorithm for fair  $k$ -median with exact fairness constraints and small  $f$ , where  $f$  is the size of a fairlet. It runs in time  $(kf)^{O(\ell)} \log n + \text{poly}(n)$ . We emphasize that even in the case  $\ell = O(1)$ , all previous results on fair representation clustering with more than two protected groups, notably [5, 8, 9], either violate the fairness constraints by additive terms or run in time that is exponential in  $k$ . In this paper, we present first polynomial time approximation algorithms for fair representation with  $\ell = O(1)$  multiple protected classes that satisfy the fairness requirement with no additive violations (see Theorem 1.3 and Theorem 1.6).

*General Representation Fairness.* Our main algorithm has three steps: *location consolidation*, *approximation the metric by a distribution of tree metrics*, and finally *solving fair clustering on a tree*.

In the first step, we run an existing constant factor approximation algorithm for  $k$ -median to find a set of  $k$  centers  $C = \{c_1, \dots, c_k\}$ . Next, we move each point to its closest center  $c_i$  in  $C$  in the constructed solution. In other words, we reduce the initial instance of size  $n$  (with possibly  $n$  different locations) to an instance of size  $n$  with exactly  $k$  locations. (i.e., we may have multiple data points mapped/moved to each location.)

In the second step, we use the metric embedding technique by Fakcharoenphol et al. [21] to approximate the reduced instance by a tree metric with expected distortion  $O(\log k)$ . As the reduced

instance, this instance also has at most  $k$  different locations. Additionally, the metric on these locations is a tree metric.

Finally, in the third step, we use dynamic programming (DP) to find a fair assignment of  $n$  points located at  $k$  different locations to  $k$  centers. The DP runs in time  $n^{O(\ell)}$ .

We note that that our first step is very similar to that used by Bera et al. [8] and Bercea et al. [9]; they first find a not-necessarily-fair clustering of the data points and then reassign the points so as to ensure that the clustering is approximately fair. In the context of  $k$ -median, the idea of approximating the input metric with a distribution of dominating trees was introduced by Bartal [7]; this approach was recently used by Backurs et al. [4] in their approximation algorithm for a different variant of fair representation clustering with 2 groups. Our dynamic programming algorithm is novel and very different from DP algorithms previously used for solving  $k$ -median on trees (see e.g., [3, 27, 40]).

**THEOREM 1.3.** *There exists a randomized  $O(\log k)$ -approximation algorithm for fair representation  $k$ -median that runs in  $n^{O(\ell)}$  time.*

Note that in practice the number of classes is usually a *small constant*. Then our algorithm runs in polynomial time. The problem is interesting even when  $\ell$  is a fixed single-digit number. In addition, note that our algorithm returns a clustering that does not violate the fairness constraints.

**REMARK 1.4.** *Our approach works in a more general setting with a set of fair profiles  $F$ , where each profile  $p \in F$  is a vector of length  $\ell$ . A cluster  $C$  (a cluster only contains one center and a  $k$ -clustering contains  $k$  clusters) is fair w.r.t.  $F$  if  $(|X_1 \cap C|, \dots, |X_\ell \cap C|) \in F$ . For example, this general notion captures the setting in which we only need to guarantee that in each cluster, a sufficiently large fraction of members belong to one of the disadvantaged groups specified by  $D \subseteq [\ell]$ ;  $\forall C, \sum_{i \in D} |X_i \cap C| \geq \alpha \cdot |C|$ . Note that our algorithm works for any given set of fair profiles  $F$ .*

*To the best of our knowledge, none of the previous results on fair representation clustering implies an approximation bound for this general “fairness profile” notion.*

*Given a clustering instance and set  $F$ , our algorithm finds a clustering such that all clusters are fair w.r.t.  $F$ . Assuming the existence of a membership oracle for  $F$  that runs in time  $t_F$ , the running time and the approximation factor are  $t_F \cdot n^{O(\ell)}$  and  $O(\log k)$  as in Theorem 1.3. In most natural scenarios the membership oracle can be implemented efficiently and the asymptotic runtime of our algorithm remains  $n^{O(\ell)}$ . For instance, in the aforementioned fairness requirement which guarantees the presence of disadvantaged groups in each cluster;  $t_F = O(\ell)$ ; hence, the total runtime of our algorithm in this setting is still  $n^{O(\ell)}$ .*

**REMARK 1.5.** *We describe and analyze our algorithm for the case when each point belongs to exactly one group  $X_i$ . However, with a minor modification, our algorithm can also handle the case when a point may belong to multiple groups or do not belong to any group. We introduce a “virtual” group  $Y_G$  for every  $G \subseteq [\ell]$  such that there exists a point that belongs to groups  $X_i$  with  $i \in G$  and only to them. Note that new virtual groups are disjoint and cover  $X$ . We define the fairness constraints as follows: for every  $j \in [\ell]$ ,  $\alpha_j |C_j| \leq \sum_{G: j \in G} |C_j \cap Y_G| \leq \beta_j |C_j|$ . By Remark 1.4, our algorithm can handle these constraints; they are equivalent to the original constraints, since*

$\sum_{G:j \in G} |C_i \cap Y_G| = |C_i \cap X_j|$ . Note that now the algorithm runs in time  $n^{O(\ell')}$  where  $\ell'$  is the number of virtual groups (clearly  $\ell' \leq 2^\ell$  but  $\ell'$  may be much smaller than  $2^\ell$ ).

*Exact Representation Fairness.* We significantly improve the running time of our algorithms when the fairness constraints are exact (see Definition 1.2) and each point belongs to exactly one group. We first run the algorithm by Bera et al. [8] that returns a set of centers and an assignment of points to these centers that “nearly” satisfies the fairness requirement. Next, we move each point to its assigned center. We prove that there exists an  $O(1)$ -approximately optimal fair assignment that only moves a set of  $O(kf^2)$  points  $S^*$ . Lastly, we show that we can find such a set  $S$  of size  $O(k^2f^2)$  in polynomial time. Then, loosely speaking, we run our main algorithm on the set  $S$ . Since  $S$  has only  $O(k^2f^2)$  points, the algorithm runs in time  $(kf)^{O(\ell)}$ .

**THEOREM 1.6.** *There exists a randomized  $O(\log k)$ -approximation algorithm for exactly fair  $k$ -median that runs in time  $(kf)^{O(\ell)} \log n + \text{poly}(n)$ , where  $f$  is the fairlet size.*

Note that our algorithm returns a clustering that does not violate the fairness constraints.

## 2 PRELIMINARIES

*Embedding into a distribution of dominating trees.* Given a set  $M$ , a metric on  $M$  is a map  $d : M \times M \rightarrow \mathbb{R}^+$  that satisfies: for all  $x, y, z \in M$ ,

- (1)  $d(x, y) = 0$  if and only if  $x = y$ ;
- (2)  $d(x, y) = d(y, x)$ ;
- (3)  $d(x, z) \leq d(x, y) + d(y, z)$ ,

where  $\mathbb{R}^+$  is the set of positive real numbers. A metric space is a pair  $(M, d)$ , where  $M$  is a set and  $d$  is a metric on  $M$ .

Let  $\chi$  be a tree on  $M$  (i.e. the vertex set of  $\chi$  is  $M$ ). Let  $E$  denote the edge set of  $\chi$  and  $w : E \rightarrow \mathbb{R}^+$  be a weight function on  $E$ . For each edge  $e \in E$ , we call  $w(e)$  the length of  $e$ . For  $u, v \in M$ , let  $P_{u,v}^\chi$  be the path from  $u$  to  $v$  in  $\chi$ . We define the shortest-path metric  $d_\chi : M \times M \rightarrow \mathbb{R}^+$  as  $d_\chi(u, v) = \sum_{e \in P_{u,v}^\chi} w(e)$ . It is easy to verify that  $d_\chi$  is a metric.

Assume that we are given a metric space  $(M, d)$ . We will consider trees  $\chi$  on  $M$  (whose edges have positive lengths) and shortest-path metrics  $d_\chi$  that they define on  $M$ . We say that  $(M, d)$  is  $\alpha$ -approximated by a probabilistic distribution  $\mathcal{D}_\chi$  of dominating trees  $\chi$  with distortion  $\alpha \geq 1$  if

- Every tree metric  $d_\chi$  in the support  $\mathcal{X}$  of  $\mathcal{D}_\chi$  dominates metric  $(M, d)$ ; i.e.,  $d_\chi(u, v) \geq d(u, v)$  for all  $u, v \in M$ .
- $\mathbf{E}_{\chi \sim \mathcal{D}_\chi} [d_\chi(u, v)] \leq \alpha \cdot d(u, v)$  for all  $u, v \in M$ .

A key component in our algorithms is the following result by Fakcharoenphol et al. [21] (see also [6] by Bartal).

**THEOREM 2.1 (FAKCHAROENPHOL ET AL. [21]).** *Every metric space  $(M, d)$  can be approximated by a distribution of dominating trees with distortion at most  $O(\log |M|)$ . Moreover, we can sample from the distribution of dominating trees in polynomial time.*

## 3 ALGORITHMS

Now we present an  $O(\log k)$ -approximation algorithm for fair representation  $k$ -median that runs in time  $n^{O(\ell)}$ , where  $n$  is the number of points and  $\ell$  is the number of groups.

In our algorithms, we are going to transform instances to “simpler” instances by moving points to new locations. It will be helpful to distinguish between data points and their locations. Our algorithms will not change data points and their memberships in groups, but will change their locations. Formally, we think of abstract data points that are mapped to points/locations in a metric space; our algorithms will modify this mapping between data points and their locations. We call this process a reassignment. We denote the set of data points by  $X$  and the set of locations by  $L$ . Initially  $L = X$  and every point  $x \in X$  is assigned to location  $x$ , but this will change when we transform the instance. We denote the location of point  $x$  w.r.t. instance  $\mathcal{I}$  by  $\text{loc}(x) = \text{loc}_{\mathcal{I}}(x)$ .

Now, if two data points at the same location belong to the same group, then they are interchangeable for our algorithm. Thus, instead of storing the actual data points, the algorithm will only store the number of points from each group at every location.

Denote  $\mathcal{R}_{\geq 0} \equiv \{0, \dots, n\}^\ell$  and  $\mathcal{R} \equiv \{-n, \dots, n\}^\ell$ . For each  $q \in L$  and  $j \in [\ell]$ , let  $v_j(q)$  be the number of data points from group  $j$  at location  $q$ . We call vector  $v(q) = (v_1(q), \dots, v_\ell(q)) \in \mathcal{R}_{\geq 0}$  the profile of location  $q$ . We define the profile of a set of data points  $S$  as a vector  $v(S)$  in  $\mathcal{R}_{\geq 0}$  whose  $j$ -th coordinate equals the number of data points from group  $j$  in the set;  $v(S) := \sum_{q \in S} v(q)$ .

Consider a set of  $k$  clusters  $C$ . To describe a clustering, we introduce an assignment function  $r : L \times C \rightarrow \mathcal{R}_{\geq 0}$ . For each center  $c \in C$ ,  $r_j(q, c)$  ( $r_j(q, c)$  is the  $j$ th entry of the vector  $r(q, c)$ ) denotes the number of points from group  $j$  at location  $q$  that we assign to  $c$ . Note that vector  $R(c) = \sum_{q \in L} r(q, c) \in \mathcal{R}_{\geq 0}$  specifies the number of points from each group that are assigned to center  $c$ . We call  $R(c)$  the profile of center  $c$ . We require that for every  $q \in L$ ,  $\sum_{c \in C} r(q, c) = v(q)$ , meaning that each data point at location  $q$  belongs to exactly one cluster.

The fairness constraints are defined by a set of fair profiles  $F \subset \mathcal{R}_{\geq 0}$ . We say that the cluster assigned to center  $c$  fairly represents groups if  $R(c) \in F$ . In fair representation clustering with fairness requirement  $\{\alpha_j, \beta_j\}_{j \in [\ell]}$

$$R(c) \in F \text{ if and only if } \alpha_j \|R(c)\|_1 \leq R_j(c) \leq \beta_j \|R(c)\|_1 \text{ for all } j \in [\ell].$$

We restate the objective of fair representation  $k$ -median as follows

$$\sum_{c \in C} \sum_{q \in L} d(q, c) \cdot \|r(q, c)\|_1. \quad (3)$$

Note that  $\|r(q, c)\|_1$  is the total number of points at location  $q$  assigned to center  $c$ .

### 3.1 From a Clustering to a New Instance

Consider an instance  $\mathcal{J}$  and a clustering for  $\mathcal{J}$  (which is not necessarily fair). As we discussed above, we can define the clustering by specifying (i) a set  $C$  of  $k$ -centers and (ii) either a mapping  $\phi$  from the set of data points  $X$  to  $C$  or, equivalently, an assignment function  $r(q, c)$ . The cost of the clustering is

$$\text{cost} = \sum_{x \in X} d(\text{loc}_{\mathcal{J}}(x), \phi(x)) = \sum_{c \in C} \sum_{q \in L} d(q, c) \cdot \|r(q, c)\|_1.$$

Let us move every data point from its original location in  $\mathcal{J}$  to  $\phi(x)$ . We get a new instance  $\mathcal{J}'$ . Note that  $\text{loc}_{\mathcal{J}'}(x) = \phi(x) \in C$ . The profile  $v'(c)$  of location  $c \in C$  is  $v'(c) = R(c)$  (the profile  $v'(x)$  of a location  $x \notin C$  is a zero vector). Instance  $\mathcal{J}'$  has the same set of fairness profiles  $F$  as the original instance  $\mathcal{J}$ .

**CLAIM 3.1.** *Consider a clustering  $(C', \phi')$  of  $X$ . Denote its cost w.r.t. instances  $\mathcal{J}$  and  $\mathcal{J}'$  by  $\text{cost}_{\mathcal{J}}$  and  $\text{cost}_{\mathcal{J}'}$ , respectively. Then  $|\text{cost}_{\mathcal{J}} - \text{cost}_{\mathcal{J}'}| \leq \text{cost}$ .*

**PROOF.** Consider a data point  $x$  and the center  $\phi'(x)$  that it is assigned to by clustering  $(C', \phi')$ . Then, by the triangle inequality,  $|d(\text{loc}_{\mathcal{J}}(x), \phi'(x)) - d(\text{loc}_{\mathcal{J}'}(x), \phi'(x))| \leq d(\text{loc}_{\mathcal{J}}(x), \text{loc}_{\mathcal{J}'}(x)) = d(\text{loc}_{\mathcal{J}}(x), \phi(x))$ . Adding up this inequality over all data points  $x$ , we get the desired inequality  $|\text{cost}_{\mathcal{J}} - \text{cost}_{\mathcal{J}'}| \leq \text{cost}$ .  $\square$

### 3.2 Location Consolidation Step

We run a constant factor approximation algorithm for the standard  $k$ -median problem on our set of data points; e.g., the algorithm by Charikar et al. [13] or by Li and Svensson [33]. We get  $k$  centers  $\bar{C} = \{\bar{c}_1, \dots, \bar{c}_k\}$  and a Voronoi assignment  $\phi$  of points to the centers.

As described above, we move each data point  $x$  to center  $\phi(x)$ . We denote the original instance by  $\mathcal{I}$  and the obtained instance by  $\mathcal{I}'$ . We will refer to  $\mathcal{I}'$  as *the reduced instance*.

**CLAIM 3.2.** *Let  $\text{OPT}_{\mathcal{I}}$  and  $\text{OPT}_{\mathcal{I}'}$  be the costs of optimal fair clusterings for  $\mathcal{I}$  and  $\mathcal{I}'$ , respectively. Assume that we used an  $\alpha$ -approximation algorithm for  $k$ -median in the consolidation step. Further assume that there is a  $\beta_k$ -approximation algorithm for fair representation  $k$ -median on  $\mathcal{I}'$ . Then, there is an  $(\alpha\beta_k + \alpha + \beta_k)$ -approximation algorithm for fair representation  $k$ -median on  $\mathcal{I}$ .*

**PROOF.** Observe that the cost of an optimal (not necessarily fair)  $k$ -median clustering of  $\mathcal{I}$  is at most the cost of an optimal fair  $k$ -median clustering of  $\mathcal{I}$ . Thus the cost of the clustering that we use in the reduction is at most  $\alpha\text{OPT}_{\mathcal{I}}$ .

Now consider the optimal fair clustering for  $\mathcal{I}$ . By Claim 3.1, the cost of this clustering as a clustering of  $\mathcal{I}'$  is at most  $(\alpha + 1)\text{OPT}_{\mathcal{I}}$ . Therefore,  $\text{OPT}_{\mathcal{I}'} \leq (\alpha + 1)\text{OPT}_{\mathcal{I}}$ .

Our approximation algorithm for  $\mathcal{I}$  is very straightforward. We simply run the  $\beta_k$ -approximation algorithm on instance  $\mathcal{I}'$  and output the obtained clustering as a clustering of  $\mathcal{I}$ . Now we upper bound the cost of this clustering w.r.t. instance  $\mathcal{I}'$  and then w.r.t.  $\mathcal{I}$ . The cost w.r.t.  $\mathcal{I}'$  is at most  $\beta_k\text{OPT}_{\mathcal{I}'} \leq \beta_k(\alpha + 1)\text{OPT}_{\mathcal{I}}$ . Consequently, by Claim 3.1, the cost w.r.t.  $\mathcal{I}$  is at most  $(\beta_k(\alpha + 1) + \alpha)\text{OPT}_{\mathcal{I}}$ , as required.  $\square$

### 3.3 Embedding into Tree Metrics

Consider restriction  $\bar{d}$  of metric  $d$  to  $\bar{C}$ . We will reduce the problem to the case when  $\bar{d}$  is a tree metric by paying a factor of  $O(\log k)$ . To this end, we will approximate  $\bar{d}$  by a distribution  $\mathcal{D}_{\chi}$  of dominating trees  $\chi$  with distortion  $O(\log k)$  and then solve the fair representation  $k$ -median for a number of trees  $\chi$  randomly sampled from  $\mathcal{D}_{\chi}$ .

**CLAIM 3.3.** *Suppose that there exists an  $\alpha$ -approximation algorithm  $\mathcal{A}$  for fair representation  $k$ -median on tree metrics. Then, there*

*is an  $O(\alpha \log k)$ -approximation algorithm for the reduced problem  $\mathcal{I}'$  that succeeds with high probability (i.e.,  $1 - n^{-c}$  for any desired constant  $c$ ).*

**PROOF.** Let  $r^*$  be the optimal fair assignment for  $\mathcal{I}'$  and  $\text{OPT}$  be its cost. Consider an approximation of  $\bar{d}$  by a distribution  $\mathcal{D}_{\chi}$  of dominating trees  $\chi$  with distortion  $\log k$ , which exists by Theorem 2.1.

For a tree metric  $d_{\chi}$ , let  $r_{\chi} : \bar{C} \times \bar{C} \rightarrow \mathcal{R}_{\geq 0}$  denote the  $O(\alpha)$ -approximate assignment of points to centers that algorithm  $\mathcal{A}$  finds on instance with metric  $d_{\chi}$ . Then,

$$\begin{aligned} & \mathbf{E}_{\chi \sim \mathcal{D}_{\chi}} \left[ \sum_{q, c \in \bar{C}} d(q, c) \cdot \|r_{\chi}(q, c)\|_1 \right] \\ & \leq \mathbf{E}_{\chi \sim \mathcal{D}_{\chi}} \left[ \sum_{w, c \in \bar{C}} d_{\chi}(q, c) \cdot \|r_{\chi}(q, c)\|_1 \right] \\ & \leq \mathbf{E}_{\chi \sim \mathcal{D}_{\chi}} \left[ \alpha \cdot \sum_{q, c \in \bar{C}} d_{\chi}(q, c) \cdot \|r^*(q, c)\|_1 \right] \\ & = \alpha \cdot \sum_{q, c \in \bar{C}} \mathbf{E}_{\chi \sim \mathcal{D}_{\chi}} [d_{\chi}(q, c)] \cdot \|r^*(q, c)\|_1 \\ & \leq \alpha \cdot \sum_{q, c \in \bar{C}} O(\log k) \cdot d(q, c) \cdot \|r^*(q, c)\|_1 \\ & = O(\alpha \log k) \cdot \text{OPT}. \end{aligned}$$

The three inequalities above hold, since (i)  $d(q, c) \leq d_{\chi}(q, c)$  (always); (ii)  $r_{\chi}$  is an  $\alpha$ -approximate assignment/solution; and (iii)  $\mathbf{E}d_{\chi}(q, c) \leq O(\log k) \cdot d(q, c)$  (recall that  $\bar{d}$  is a restriction of  $d$  to  $\bar{C}$ ; for every  $u, v \in \bar{C}$ ,  $\bar{d}(u, v) = d(u, v)$ ).

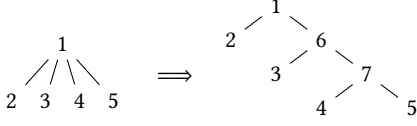
By Markov's inequality the obtained solution  $r_{\chi}$  approximates the optimal assignment within a factor of  $O(\alpha \log k)$  with probability at least  $1/2$ . By running the proposed algorithm  $\Theta(\log n)$  times, we get an  $O(\alpha \log k)$ -approximate solution w.h.p.  $\square$

### 3.4 Reduced Assignment Problem on Trees

In this section, we assume that  $(L, d)$  is a tree metric on  $k$  points with profile vector  $v(u) \in \mathcal{R}_{\geq 0}$  for every location  $u \in L$ . We open a center at every location and our goal now is to find a fair assignment of data points to centers. Recall that our notion of fairness is more general than that in Definition 1.1 (we discussed it in Remark 1.4).

**3.4.1 Conversion to Binary Tree.** We choose an arbitrary root in the tree. It is convenient now to convert the tree to a *binary* tree in which every non-leaf vertex has exactly two children. We do that by adding *Steiner locations*. Namely, we replace each vertex  $u$  with  $k > 2$  children with a path of length  $k - 2$ . The first vertex on the path is  $u$  (we assume  $u_0 \equiv u$ ); other vertices  $u_1, \dots, u_{k-2}$  are new Steiner locations. We keep the parent of  $u$  unchanged; we let the parent of  $u_i$  be  $u_{i-1}$  (for  $i \geq 1$ ). For  $j \in \{2, \dots, k-1\}$ , we reassign  $j$ -th child  $v_j$  of  $u$  to  $u_{j-1}$  and reassign  $k$ -th child  $v_k$  to  $u_{k-2}$ . We set the length of all edges on the path  $u \rightarrow u_1 \rightarrow \dots \rightarrow u_{k-2}$  to 0; we let the length of each edge  $(u_{j-1}, v_j)$  be that of  $(u, v_j)$  in the original tree; the length of  $(u_{k-2}, v_k)$  to that of  $(u, v_k)$  in the original tree. Additionally, we add a Steiner child to vertices that have exactly one child. For a concrete example, consider a tree rooted at vertex

1, in which 1 has four children 2, 3, 4, 5. We transform the tree by adding Steiner nodes 6 and 7 as follows:



Let  $A$  be the set of non-Steiner locations and  $B$  be the set of Steiner locations. We open a center at every  $a \in A$ ; we do not open centers at any Steiner location  $b \in B$ . For  $b \in B$ , let  $v(b) = 0$ .

**3.4.2 Dynamic Program.** Now we write a dynamic program (DP). We first recall the setup. We are given a binary tree  $L$ , which contains Steiner and non-Steiner nodes/locations. Each non-Steiner node contains some data points that we want to cluster and each Steiner node contains no data points. Our goal is to move the data points around (assign data points to the non-Steiner nodes) such that the resulting clustering is fair. The objective is to minimize the “assignment cost”  $\sum_{t,c \in A} d(t,c) \cdot \|r(t,c)\|_1$ , where  $\|r(t,c)\|_1$  is the number of data points at location  $t$  that are assigned to  $c$ .

Let  $T_u$  be the subtree of  $L$  rooted at  $u$ . Let  $A_u$  and  $B_u$  be non-Steiner and Steiner locations in  $T_u$ , respectively. For a fixed assignment  $\phi$ , data points located at nodes/locations in  $T_u$  can be classified into two types: *local points* and *out-points*. A *local point* is assigned to a node/location in  $T_u$  by  $\phi$  and an *out-point* is assigned to a node/location outside  $T_u$  by  $\phi$ . In addition, there are some points outside  $T_u$  that are assigned to nodes/locations in  $T_u$ . We refer to these points as *in-points*. Now we define two functions  $\rho_{out} : A_u \rightarrow \mathcal{R}_{\geq 0}$  and  $\rho_{in} : A_u \rightarrow \mathcal{R}_{\geq 0}$ , which specify the out-points and in-points, respectively. Namely, for each location  $c \in A_u$ ,  $\rho_{out}(c)$  is the profile of out-points located at  $c$ , and  $\rho_{in}(c)$  is the profile of in-points assigned to  $c$ . Let  $q := \sum_{c \in A_u} \rho_{in}(c) - \sum_{y \in A_u} \rho_{out}(y)$  be the “net-imports” of  $T_u$ . Clearly,  $q \in \mathcal{R}$ .

Now, we create a DP-table  $M[u, q]$  where  $u \in L$  and  $q \in \mathcal{R}$ . Loosely speaking,  $M[u, q]$  is the cost of the minimum cost partial solution such that (a) clusters for all centers  $c \in A_u$  satisfy fairness constraints (note that each cluster consists of the points at locations in  $T_u$  assigned to  $c$  and in-points assigned to  $c$ ) and (b) the difference between the number of in-points and out-points from group  $j$  equals  $q_j$ . The cost of a partial solution comprises (i) the assignment costs for points in  $T_u$  assigned to centers in  $T_u$ , (ii) for each out-point  $x$  located at  $y \in A_u$ , the portion  $d(y, u)$  of the assignment cost of  $x$  that “lies” inside  $T_u$ , and (iii) for each in-point  $x'$  assigned to center  $c \in A_u$ , the portion  $d(u, c)$  of the assignment cost of  $x'$  that “lies” inside  $T_u$ .

Formally,  $M[u, q]$  is the cost of the optimal solution for the following problem.

*Find.*

- function  $r : A_u \times A_u \rightarrow \mathcal{R}_{\geq 0}$ ; map  $r$  specifies the assignment of data points located in subtree  $T_u$  to centers in  $T_u$ . Namely,  $r(y, c)$  is the profile of the set of data points located at  $y$  that are assigned to center  $c$ . Let  $R_u(c) = \sum_{y \in T_u} r(y, c)$ . Note that  $R_u(c)$  is the profile of the set of data points at locations in  $A_u$  that are assigned to center  $c$ .
- function  $\rho_{out} : A_u \rightarrow \mathcal{R}_{\geq 0}$ ; map  $\rho_{out}$  specifies the set of data points in subtree  $T_u$  that are not assigned to any center

in  $T_u$ ; these data points will have to be assigned to centers outside of  $T_u$  later. Namely,  $\rho_{out}(y)$  is the profile of currently unassigned data points at location  $y$ .

- function  $\rho_{in} : A_u \rightarrow \mathcal{R}_{\geq 0}$ ;  $\rho_{in}$  specifies how many data points located outside of the subtree  $T_u$  are assigned to centers in  $T_u$ . Namely,  $\rho_{in}(c)$  is the profile of data points that lie outside of  $T_u$  but assigned to center  $c$ .

*Such that.*

- for every  $y \in A_u$ ,  $\sum_{c \in A_u} r(y, c) + \rho_{out}(y) = v(y)$ . This condition says that every data point at location  $y$  is assigned to some center  $c \in A_u$  or is currently unassigned.
- $R_u(c) + \rho_{in}(c) \in F$  for all  $c \in A_u$ . This condition says that the profile of the cluster centered at  $c$  is in  $F$  (that is, the cluster satisfies the fairness constraints). Here,  $R_u(c)$  counts data points in  $T_u$  that are assigned to  $c$  and  $\rho_{in}(c)$  counts data points outside of  $T_u$  that are assigned to  $c$ .
- $\sum_{c \in A_u} \rho_{in}(c) - \sum_{y \in A_u} \rho_{out}(y) = q$ .

*Cost.* The objective is to minimize the following cost

$$\sum_{y,c \in A_u} d(y,c) \cdot \|r(y,c)\|_1 + \sum_{c \in A_u} d(c,u) \cdot \|\rho_{in}(c)\|_1 + \sum_{y \in A_u} d(y,u) \cdot \|\rho_{out}(y)\|_1.$$

If there is no feasible solution the cost is  $+\infty$ .

**3.4.3 Solving the DP.** We fill out the DP table starting from the leaves and going up to the root (bottom-up).

Computing  $M[u, q]$  for leaves  $u$  is straightforward using the definition of  $M[u, q]$ . In this case, the subtree  $T_u$  rooted at  $u$  is a single node. Thus, for any given  $q$ , the profile at  $u$  after assignment is

$$R_u(u) + \rho_{in}(u) = r(u, u) + \rho_{in}(u) = r(u, u) + \rho_{out}(u) + q = v(u) + q.$$

If  $u$  is not a Steiner node, then  $M[u, q] = 0$  if  $v(u) + q \in F$  and  $M[u, q] = \infty$  if  $v(u) + q \notin F$ . If  $u$  is a Steiner node, then  $M[u, q] = 0$  if  $q = 0$  and  $M[u, q] = \infty$  if  $q \neq 0$ .

Now assume that  $u$  is not a leaf; since the tree is a binary tree,  $u$  has two children  $y$  and  $z$ . Note that node  $u$  will send  $q_y$  points to  $T_y$  and  $q_z$  points to  $T_z$  for some  $q_y, q_z \in \mathcal{R}$ . Then  $M[u, q]$  is the minimum over  $q_y, q_z \in \mathcal{R}$  satisfying

- if  $u \in A$ , then  $v(u) + q - q_y - q_z \in F$ ; that is, cluster centered at  $u$  satisfies fairness constraints.
- if  $u \in B$ , then  $q - q_y - q_z = 0$ , since  $v(u) = 0$  and no point will be assigned to  $u$ .

of the following cost

$$M[y, q_y] + M[z, q_z] + d(u, y) \cdot \|q_y\|_1 + d(u, z) \cdot \|q_z\|_1 \quad (4)$$

The pseudo-code of the DP algorithm is provided in Algorithm 1.

**LEMMA 3.4.** *The described dynamic programming algorithm runs in time  $n^{O(\ell)}$  and finds an optimal fair assignment of the data points  $X$  located in the set of locations  $L$ .*

**PROOF.** It is straightforward that the DP algorithm correctly computes the DP entries. The cost of the optimal fair assignment for our problem is given by  $M[\text{root}, \mathbf{0}]$  where  $\text{root}$  is root of the tree and  $\mathbf{0}$  is the all-zero vector.

The update rule, which is specified by Eq. (4) and the constraints on  $q_y, q_z, q$  and  $v(u)$ , computes  $M[u, q]$  correctly in time  $n^{O(\ell)}$ , which accounts for enumerating over all possible values of  $q_y$  and  $q_z$ . Since the DP tables contains  $k \cdot n^{O(\ell)}$  cells, the total time to fully compute the table is  $k \cdot n^{O(\ell)} = n^{O(\ell)}$ . Finally, once the whole table is computed, we can recover the assignment itself by traversing the table from  $M[\text{root}, 0]$  in the reverse direction, which takes  $n^{O(\ell)}$  time. The total running time is  $n^{O(\ell)}$ .  $\square$

**PROOF OF THEOREM 1.3:** We first apply the consolidation step and thus reduce the problem to the set of locations  $\bar{C}$  is of size  $k$ . Then we approximate metric on  $\bar{C}$  by a distribution of dominating tree metrics, as explained in 3.3. We obtain a logarithmic number of instances with tree metrics. We exactly solve each of them using the DP algorithm described above. Finally, we return the best of the clusterings we find. It follows from Claims 3.2 and 3.3 and Lemma 3.4, that the algorithm finds an  $O(\log k)$  approximation with high probability in time  $n^{O(\ell)}$ .  $\square$

---

**Algorithm 1** DP Algorithm on Trees
 

---

**Require:**  $T$  is a binary tree of height  $d$ ,  $r$  is the root of  $T$ , and for each  $i = 0, 1, \dots, d$ ,  $N_i$  is the set of nodes in the  $i$ -th level of  $T$ .

```

1: for  $u \in N_d$  do
2:   for  $q \in \mathcal{R}$  do
3:     if  $u$  is a non-Steiner node then
4:       if  $v(u) + q \in F$  then
5:          $M[u, q] = 0$ 
6:       else
7:          $M[u, q] = \infty$ 
8:       end if
9:     else
10:      if  $q = 0$  then
11:         $M[u, q] = 0$ 
12:      else
13:         $M[u, q] = \infty$ 
14:      end if
15:    end if
16:  end for
17: end for
18: for  $i = d - 1, d - 2, \dots, 0$  do
19:   for  $u \in N_i$  do
20:      $y, z \leftarrow$  children of  $u$ 
21:     if  $u$  is a Steiner node then
22:        $M[u, q] = \min\{M[y, q_y] + M[z, q_z] + d(u, y) \cdot \|q_y\|_1 +$ 
23:          $d(u, z) \cdot \|q_z\|_1 : q_y, q_z \in \mathcal{R}, q = q_y + q_z\}$ 
24:     else
25:        $M[u, q] = \min\{M[y, q_y] + M[z, q_z] + d(u, y) \cdot \|q_y\|_1 +$ 
26:          $d(u, z) \cdot \|q_z\|_1 : q_y, q_z \in \mathcal{R}, v(u) + q - q_y - q_z \in F\}$ 
27:     end if
28:   end for
29: end for
30: return  $M[r, 0]$ 

```

---

## 4 SPECIAL CASE OF EXACT FAIRNESS

In this section, we design an  $O(\log k)$  approximation algorithm for  $k$ -median with exact fairness constraints. The algorithm runs in time  $(kf)^{O(\ell)} \log n + \text{poly}(n)$  and succeeds with high probability.

To recall, we say that a cluster centered at  $c$  is *exactly fair* if  $\forall j \in [\ell], R_j(c) = \frac{|X_j|}{|X|} \|R(c)\|_1$ , where  $\frac{|X_j|}{|X|}$  is the proportion of data points in  $X$  that belongs to group  $j$ . We say that a clustering is *exactly fair* if all of its clusters are exactly fair. More generally, a set of data points  $S$  with profile  $v(S) \in \mathcal{R}_{\geq 0}$  is *exactly fair* if for all  $j \in [\ell], v_j(S) = \frac{|X_j|}{|X|} \|v(S)\|_1$ .

Recall that we defined *fairlet* in Definition 1.2 as a minimal size non-empty set that is exactly fair. Note that all fairlets have the same size and we use  $f$  to denote their size. Further, for every  $j \in [\ell]$ , we use  $f_j$  to denote the number of points from group  $j$  in any fairlet.

### 4.1 Reassignment Method

We start with the reduced instance  $I'$  on the set of locations  $\bar{C}$ , which we constructed in the location consolidation step.

Given any subset  $S \subseteq X$  with profile vector  $v(S)$ , we say that  $v(S)$  is  $\gamma$ -approximately fair if

$$\left| v_j(S) - \frac{f_j}{f} \|v(S)\|_1 \right| \leq \gamma, \quad \forall j \in [\ell]. \quad (5)$$

Given an assignment  $r$ , we say that  $r$  is a  $\gamma$ -approximately fair assignment if  $R(c)$  is  $\gamma$ -approximately fair for all  $c \in C$ , where  $R(c) = \sum_{x \in X} r(x, c) \in \mathcal{R}_{\geq 0}$  is the center profiles corresponding to  $r$ .

We use the following result by Bera et al. [8].

**THEOREM 4.1.** *For any metric space  $X$  and a set of centers  $C \subseteq X$ , There exists a 3-approximately fair assignment  $r : X \times C \rightarrow \mathcal{R}_{\geq 0}$  whose cost is no more than the optimal fair assignment of  $X$ . Moreover, there is an algorithm which finds this assignment in polynomial time.*

We start by running the assignment algorithm from Theorem 4.1 on instance  $I'$ . Let  $r$  denote this new assignment and  $R(c) = \sum_{x \in X} r(x, c) \in \mathcal{R}_{\geq 0}$  be the center profiles corresponding to  $r$ . As discussed in Section 3.1, every assignment defines a new instance. Let  $I''$  be the instance defined by assignment  $r$ . Now our goal is to modify  $r$  by reassigning only a small number of points so that the obtained assignment is fair. We describe this reassigning procedure below.

**LEMMA 4.2.** *Assume that there is a  $\beta_k$ -approximation algorithm for the fair clustering problem on  $I''$ . Then, there is an  $(2\beta_k + 1)$ -approximation algorithm for  $I'$ .*

**PROOF.** The proof is similar to that of Claim 3.2. We run the  $\beta_k$  approximation algorithm on instance  $I''$  and output the obtained clustering as a clustering for  $I'$ . We now upper bound its cost.

Let  $\text{OPT}_{I'}$  and  $\text{OPT}_{I''}$  be the costs of the optimal fair clusterings/assignments for instances  $I'$  and  $I''$ , respectively. By Claim 3.1 applied to  $r$ ,  $\text{OPT}_{I''} \leq 2\text{OPT}_{I'}$ . Thus the cost of the clustering we find w.r.t.  $I''$  is at most  $2\beta_k \text{OPT}_{I'}$ . Now applying Claim 3.1 to this clustering, we get that its cost w.r.t.  $I'$  is at most  $(2\beta_k + 1)\text{OPT}_{I'}$ , as required.  $\square$

*Algorithm for the reassignment problem.* We present an  $O(\log k)$ -approximation algorithm for the reassignment problem that runs in time  $(fk)^{O(\ell)}$ .

Suppose we are given an instance  $\mathcal{J}$  with locations  $c_1, \dots, c_k$ , metric  $d$  and profile vectors  $v(c_1), \dots, v(c_k)$ , that satisfy 3-approximately fair requirement as in Eq. (5). We will apply our reassignment algorithm to  $\mathcal{J} = \mathcal{I}''$ . For every  $i \in [k]$ , let  $C_i$  be the set of points assigned to  $c_i$  in  $\mathcal{J}$ .

**Definition 4.3.** For every  $i \in [k]$ , let  $F_i \subseteq C_i$  be a maximal exactly fair subset of  $C_i$ . Decompose each  $F_i$  into  $n_i = |F_i|/f$  fairlets  $\mathcal{F}_1^{(i)}, \dots, \mathcal{F}_{n_i}^{(i)}$ . Let  $P_i = C_i \setminus F_i$  be the set of remaining points. We say that  $\mathcal{D} = \{(\mathcal{F}_1^{(i)}, \dots, \mathcal{F}_{n_i}^{(i)}, P_i)\}_{i \in [k]}$  is a fairlet decomposition for clustering  $C := (C_1, \dots, C_k)$ . We call points in  $P_i$  problematic points.

**LEMMA 4.4.** Consider a fairlet decomposition for  $C$ . Then for every  $i \in [k]$ ,  $|P_i| < 4f$ .

**PROOF.** Consider an arbitrary  $i \in [k]$ . We first assume that  $n_i = 0$ ; in other words,  $F_i$  is empty. Let  $v(C_i)$  be the profile vector of  $C_i$ . Since  $F_i = \emptyset$ ,  $P_i = C_i$  and there exists a group  $j \in [\ell]$  such that  $v_j(C_i) < f_j$ . Further, since  $P_i = C_i$  is 3-approximately fair,  $|v_j(C_i) - \frac{f_j |P_i|}{f}| \leq 3$ , which implies that  $\frac{f_j |P_i|}{f} \leq v_j(C_i) + 3 < f_j + 3$ . Dividing both sides of the above equation by  $f_j$ ,  $\frac{|P_i|}{f} < \frac{3}{f_j} + 1 \leq 3 + 1 = 4$ , since  $f_j \geq 1$ . Thus,  $|P_i| < 4f$ .

Now, we drop the condition  $n_i = 0$ . Let  $v(P_i)$  be the profile vector for  $P_i$ . We show that not only  $v(C_i)$  but also  $v(P_i)$  satisfies Eq. (5) with  $\gamma = 3$ . Indeed, for each  $j \in [\ell]$ ,  $v_j(C_i) - v_j(P_i) = n_i f \frac{f_j}{f} = (\|v(C_i)\|_1 - \|v(P_i)\|_1) \frac{f_j}{f}$ . Thus, for each  $j \in [\ell]$ ,  $|v_j(P_i) - \frac{f_j}{f} \|v(P_i)\|_1| = |v_j(C_i) - \frac{f_j}{f} \|v(C_i)\|_1| \leq 3$ . Hence the same argument as for the case  $n_i = 0$  shows that  $|P_i| < 4f$ .  $\square$

Next, we give an outline of our approach.

- (1) First, we show that there exists an exactly fair assignment  $r$  for  $\mathcal{J}$  of cost at most  $2\text{OPT}_{\mathcal{J}}$  that only reassigns  $O(kf^2)$  points.
- (2) Next, we show that it is possible to find a subset  $S_i \subseteq C_i$  for each  $i \in [k]$  such that  $|S_i| = O(kf^2)$  for all  $i \in [k]$ , and there exists a solution of cost at most  $2\text{OPT}_{\mathcal{J}}$  that only reassigns points in  $S := \bigcup_{i \in [k]} S_i$ .
- (3) Finally, we apply our dynamic programming approach to reassign points in  $S$ . Since  $|S| = O(k^2 f^2)$ , the DP algorithm runs in  $(kf)^{O(\ell)}$  time.

To show (1), we impose an additional assumption on the structure of the desired (re)assignment  $r$ . Let  $C'_1, \dots, C'_k$  be the clustering resulted from an exactly fair assignment of  $\mathcal{J}$ , where  $C'_i$  is the cluster of points at location  $c_i$  after the reassignment.

Let  $C_1, \dots, C_k$  be the clustering before the (re)assignment. Now, we construct fairlet decompositions for  $(C_1, \dots, C_k)$  and  $(C'_1, \dots, C'_k)$ . Let  $E \subseteq X$  be the set of data points that are assigned to the same center before and after (re)assignment  $r$ . For each  $i \in [k]$ , let  $E_i = \{x \in E : \text{loc}_{\mathcal{J}}(x) = c_i\}$  be the set of data points in  $C_i$  that are fixed by  $r$ .

For every  $i \in [k]$ , let  $F'_i \subseteq E_i$  be a maximal exactly fair subset of  $E_i$ . We decompose  $F'_i$  into  $n'_i = |F'_i|/f$  fairlets. We call this

decomposition  $\mathcal{D}_E$ . Now, we greedily extend  $\mathcal{D}_E$  to a fairlet decomposition for  $(C_1, \dots, C_k)$ . Let  $P_i$  be the set of problematic points in the obtained decomposition. Note that Lemma 4.4 applies to  $P_i$  and therefore  $|P_i| < 4f$ .

Similarly, we greedily extend decomposition  $\mathcal{D}_E$  to a fairlet decomposition for  $(C'_1, \dots, C'_k)$ . We refer to fairlets that we add to  $\mathcal{D}_E$  as *new fairlets*. Let  $\mathcal{N}$  be the set of all new fairlets.

**Definition 4.5 (restricted assignment).** In the restricted assignment problem, we need to find a minimum cost assignment  $r : \{c_1, \dots, c_k\} \times [k] \rightarrow \mathcal{R}_{\geq 0}$  of  $\mathcal{J}$  such that

- $r$  is a valid exactly fair assignment. Formally,

$$\begin{aligned} \forall c \in \{c_1, \dots, c_k\}, \quad & \sum_{i=1}^k r(c, c_i) = v(c) \\ \forall j \in [\ell], c \in \{c_1, \dots, c_k\}, \quad & R_j(c) = \frac{f_j}{f} \|R(c)\|_1 \end{aligned}$$

- every new fairlet  $\mathcal{F}$  in  $C'_i$  contains at least one point from  $C_i$ .

**LEMMA 4.6.** Let  $\text{OPT}_{\mathcal{J}}^R$  and  $\text{OPT}_{\mathcal{J}}$  denote the optimal values of the restricted and vanilla fair assignment problems with input  $\mathcal{J}$  respectively. Then,  $\text{OPT}_{\mathcal{J}}^R \leq 2\text{OPT}_{\mathcal{J}}$ .

**PROOF.** Let  $r$  be an optimal fair assignment of  $\mathcal{J}$ . Let  $C'_1, \dots, C'_k$  be the resulting clusters; the set  $C'_i$  consists of the points assigned to  $c_i$  by  $r$ . Now, suppose that there exists a new fairlet  $\mathcal{F}$  in some  $C'_i$  ( $i \in [k]$ ) such that  $\mathcal{F} \cap C_i = \emptyset$ . Without loss of generality, assume that the points in  $\mathcal{F}$  are from  $C_{j_1}, C_{j_2}, \dots, C_{j_m}$  for some  $m \in [k]$ . Let  $w_t := |C_{j_t} \cap \mathcal{F}|$  for every  $t \in [m]$  and  $W = \sum_{t=1}^m w_t$ . Then, the total cost of forming  $\mathcal{F}$  is  $\sum_{t=1}^m w_t \cdot d(c_{j_t}, c_i)$ . Now, for every  $s \in [m]$ , compute the cost of moving all points in fairlet  $\mathcal{F}$  to cluster  $C_{j_s}$  (i.e., reassign all data points in  $\mathcal{F}$  to center  $c_{j_s}$ ) and denote it as  $M_s$ .

$$\begin{aligned} M_s &= \sum_{t=1}^m w_t d(c_{j_t}, c_{j_s}) \\ &\leq \sum_{t \in [m] \setminus \{s\}} w_t (d(c_{j_t}, c_i) + d(c_{j_s}, c_i)) \\ &= \left( \sum_{t \in [m] \setminus \{s\}} w_t \right) d(c_{j_s}, c_i) + \sum_{t \in [m] \setminus \{s\}} w_t d(c_{j_t}, c_i). \end{aligned}$$

Next, consider the convex combination  $\sum_{s=1}^m \frac{w_s}{W} M_s$ .

$$\begin{aligned} & \sum_{s=1}^m \frac{w_s}{W} M_s \\ &= \sum_{s=1}^m \frac{w_s}{W} \sum_{t=1}^m w_t d(c_{j_t}, c_{j_s}) \\ &\leq \sum_{s=1}^m \frac{w_s}{W} \left( \sum_{t \in [m] \setminus \{s\}} w_t \right) d(c_{j_s}, c_i) + \sum_{s=1}^m \frac{w_s}{W} \sum_{t \in [m] \setminus \{s\}} w_t d(c_{j_t}, c_i) \\ &< 2 \sum_{t=1}^m w_t d(c_{j_t}, c_i). \end{aligned}$$

This implies that there exist  $s^* \in [m]$  with  $C_{j_{s^*}} \cap \mathcal{F} \neq \emptyset$  such that the cost  $M_{s^*}$  of reassigning  $\mathcal{F}$  to  $c_{j_{s^*}}$  is at most twice the current assignment cost of  $\mathcal{F}$ .

We perform this fairlet reassignment procedure to all new fairlets that do not satisfy the restricted assignment property and obtain a new restricted assignment  $r'$ . Our argument shows that the assignment cost of  $r'$  is at most twice the assignment cost of  $r$ ; hence,

$$\text{OPT}_{\mathcal{J}}^R \leq 2\text{OPT}_{\mathcal{J}}.$$

□

Next, we show that the optimal restricted assignment is even more structured.

LEMMA 4.7. *There exists an optimal solution  $r'$  to the restricted assignment of  $\mathcal{J}$  such that*

- (a) *for any  $i \in [k]$  and any new fairlet  $\mathcal{F} \subseteq C'_i$ , there exists a problematic point  $x \in \mathcal{F} \cap C_i$ ;*
- (b) *if  $r'$  moves a point  $x \in \mathcal{F}'$  for some fairlet  $\mathcal{F}' \subseteq C_i$  and  $i \in [k]$ , then  $r'$  moves all points in  $\mathcal{F}'$ .*

PROOF. Let  $r$  be the optimal solution for the restricted assignment problem. We apply two transformations to it.

*Transformation I: Simplifying the reassignment multigraph.* For every group  $j \in [\ell]$ , define the reassignment multigraph  $G_j$  on the set of centers  $c_1, \dots, c_k$ . There is a directed edge from  $c_a$  to  $c_b \neq c_a$  in  $G_j$  for every point  $x$  that is reassigned from  $c_a$  to  $c_b$  by  $r$ . We label such an edge with label  $x$ . Note that there are  $r_j(c_a, c_b)$  parallel edges from  $c_a$  to  $c_b$  in  $G_j$ .

Assume that there is a vertex  $c$  that has both incoming and outgoing edges in  $G_j$ . Let  $(c', c)$  be an incoming edge and  $(c, c'')$  be an outgoing edge (it is possible that  $c' = c''$ ). Denote their labels by  $x$  and  $x'$ . By the definition of the graph,  $r$  reassigns point  $x \in X_j$  from  $c'$  to  $c$  and point  $x' \in X_j$  from  $c$  to  $c''$ . We modify  $r$  as follows (see Figure 1). We only change the assignment of points  $x$  and  $x'$ : we reassign  $x$  to  $c''$  and  $x'$  to  $c$ . Denote the obtained assignment  $\tilde{r}$ . Since  $x$  and  $x''$  belong to the same group  $X_j$ , we do not change the profiles of any clusters and thus  $\tilde{r}$  is exactly fair. Further, if  $r$  assigns a point  $y$  from  $c_i$  to  $c_i$  then so does  $\tilde{r}$ . Therefore,  $\tilde{r}$  is also a solution to the restricted assignment problem. Finally, its cost is at most that of  $r$ : indeed the assignment costs of all points other than  $x$  and  $x'$  do not change; the total assignment cost of  $x$  and  $x'$  was equal to  $d(c', c) + d(c, c'')$  for  $r$  and is equal to  $d(c', c'')$  for  $\tilde{r}$ ; we have  $d(c', c) + d(c, c'') \geq d(c', c'')$  by the triangle inequality.

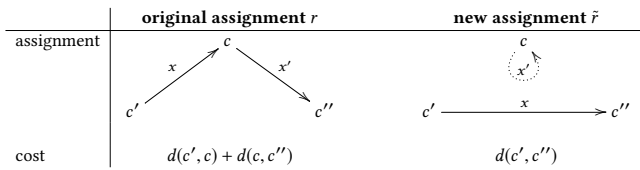


Figure 1: We reassign  $x$  and  $x'$ . When we do the reassignment, we replace two edges  $(c', c)$  and  $(c, c'')$  with one edge  $(c', c'')$  in graph  $G_j$ . Note that there are no loops in  $G_j$ . We have drawn a dotted arrow from  $c$  to itself in the figure simply to indicate that  $\tilde{r}$  does not reassign  $x'$ .

We perform this step over and over until there are no vertices that simultaneously have incoming and outgoing edges. (Note that each time we perform this step, the number of edges in one of the graphs  $G_j$  decreases by 1 and does not change in all other graphs  $G_{j'}$  (for every  $j' \in [\ell] \setminus \{j\}$ ). So we will necessarily stop at some point.)

Denote the obtained assignment by  $r'$ . It is a fair assignment and its cost is at most that of  $r$ . Consider the reassignment multigraphs  $\{G'_j\}_{j \in [\ell]}$  for  $r'$ . Each vertex  $c$  in  $G'_j$  is either a source, sink, or isolated vertex.

*Transformation II: Specifying which points  $r'$  moves.* We go over all  $i \in [k]$  and  $j \in [\ell]$  such that  $c_i$  is a source-vertex in  $G'_j$ . Assignment  $r'$  moves  $M = \sum_{i' \neq i} r_j(c_i, c_{i'})$  points in group  $j$  from  $C_i$  to other clusters. Since all points in  $C_i \cap X_j$  are interchangeable, we are free to choose any subset of  $M$  points in  $C_i \cap X_j$  to move. We choose this set in a special way. Let  $\mathcal{F}_1, \dots, \mathcal{F}_{n_i}$  be fairlets in the fairlet decomposition we constructed for  $C_i$ . We order all points in  $C_i \cap X_j$  as follows: first points from  $P_i \cap X_j$ , then from  $\mathcal{F}_1 \cap X_j$ , then from  $\mathcal{F}_2 \cap X_j$ , then from  $\mathcal{F}_3 \cap X_j$ , etc (we order points inside each of these sets arbitrarily). We choose the subset of the first  $M$  points w.r.t. this order and assume that  $r'$  moves them. Note that  $r'$  moves non-problematic points from  $C_i \cap X_j$  to another cluster only if it moves all problematic points from  $C_i \cap X_j$ .

We have defined  $r'$ . Now we prove that it satisfies properties (a) and (b). To this end, we consider a cluster  $C_i$  and analyze the following two cases.

*Case 1: Assignment  $r'$  does not move any non-problematic points from cluster  $C_i$  to other clusters. All points (if any) it moves from  $C_i$  to other clusters are problematic.* Then  $C_i \setminus P_i \subseteq C'_i$  and thus all fairlets present in the decomposition for  $C_i$  are also present in that for  $C'_i$ . Hence, every non-problematic point belongs to the same fairlet in  $C_i$  and  $C'_i$ . In particular, only problematic points may belong to new fairlets. On the other hand, since  $r'$  is a solution to the restricted assignment problem, at least one point  $x$  in each new fairlet  $\mathcal{F}$  in  $C'_i$  must be from  $C_i$ . It follows that this point must be problematic. We proved that property (a) holds in this case. Since  $r$  does not move any points from fairlets  $\mathcal{F}' \subseteq C_i$  to other clusters, property (b) trivially holds.

*Case 2: Assignment  $r'$  moves at least one non-problematic point from  $C_i$  to another cluster.* Let  $t$  be maximal index such that  $r'$  moves points from fairlet  $\mathcal{F}_t$  to other clusters. Since we are guaranteed that  $r'$  moves at least one non-problematic point,  $t$  is well defined.

By our choice of  $t$ ,  $r'$  does not move any points from fairlets  $\mathcal{F}_{t+1}, \dots, \mathcal{F}_{n_i}$  and therefore these fairlets are still present in  $C'_i$ . Now we show that there are no other fairlets in  $C'_i$ ; in particular, there are no new fairlets.

Consider a point  $x$  in  $\mathcal{F}_t$  that  $r'$  moves to some other cluster  $C_{i'}$ . Assume that  $x \in X_j$ . Then edge  $(c_i, c_{i'})$  – an edge outgoing from  $c_i$  – is present in  $G'_j$ . Hence,  $c_i$  cannot be a sink or isolated vertex and must be a source-vertex in  $G'_j$ . Therefore,  $r'$  does not move any point  $y \in X_j$  from another cluster to  $C_i$ . Also, all points in  $C_i \cap X_j$  that precede  $x$  w.r.t. the order we considered in the Transformation II step are moved to other clusters by  $r'$ ; in particular, all points in  $P_i \cap X_j, \mathcal{F}_1 \cap X_j, \dots, \mathcal{F}_{t-1} \cap X_j$  as well as  $x$  itself are moved to other clusters by  $r'$ . Therefore,  $C'_i \cap X_j$  consists of points from



$\mathcal{F}_{t+1} \cap X_j, \dots, \mathcal{F}_{n_i} \cap X_j$  and hypothetically some points from  $\mathcal{F}_t \cap X_j$ . However, point  $x$  from  $\mathcal{F}_t \cap X_j$  is assigned to another cluster and thus there are not enough points from group  $j$  left in  $\mathcal{F}_t$  to form another fairlet. We conclude that  $\mathcal{F}_{t+1}, \dots, \mathcal{F}_{n_i}$  are the only fairlets in  $C'_i$ . Since  $C'_i$  satisfies the exact fairness constraints, each point in  $C'_i$  lies in a fairlet, that is, in one of these fairlets. Now (a) trivially holds since there are no new fairlets in  $C'$ ; (b) holds since  $r'$  moves all points from fairlets  $\mathcal{F}_1, \dots, \mathcal{F}_t$  and no points from fairlets  $\mathcal{F}_{t+1}, \dots, \mathcal{F}_{n_i}$ .  $\square$

Lemma 4.7 immediately implies the following result which proves 1 and 2.

**COROLLARY 4.8.** *Suppose that  $\mathcal{J}$  is a 3-approximately fair instance. There exists an assignment  $r'$  of cost at most  $2\text{OPT}_{\mathcal{J}}$  that only moves  $O(kf^2)$  points. Furthermore, we can choose a subset  $S$  of data points such that  $|S| = O(k^2f^2)$  and the solution  $r'$  only moves points in  $S$ .*

**PROOF.** Let  $r'$  be an optimal restricted assignment that satisfies both conditions in Lemma 4.7. In the clustering defined by  $r'$ , every new fairlet contains a problematic point from the original cluster of the center they are assigned to by  $r'$ . Since by Lemma 4.4 there are only  $4kf$  problematic points in total, there are at most  $4kf$  many new fairlets. Thus,  $r'$  moves at most  $4kf^2$  many points. Moreover, by Lemma 4.6, the cost of  $r'$  is at most  $2\text{OPT}_{\mathcal{J}}$ .

Since we know that  $r'$  moves at most  $4kf^2$  many points and  $r'$  satisfies condition (b) in Lemma 4.7,  $r'$  would move at most  $4kf$  fairlets in each cluster. Recall that each  $C_i$  contains  $n_i$  fairlets. From each cluster  $C_i$ , we pick  $\min(n_i, 4kf)$  many fairlets and add them to the set  $S$ . Then, we add all the problematic points to  $S$ . Then,  $|S| \leq 4k^2f^2 + 4kf$  and we can assume that  $r'$  only moves the points in  $S$ .  $\square$

Now, we can solve the assignment problem by passing the set  $S$  as input to our dynamic programming approach described in Section 3.4.

**PROOF OF THEOREM 1.6.** Let  $\mathcal{I}$  be an instance of the exactly fair  $k$ -median problem. First, we perform location consolidation on  $\mathcal{I}$  to obtain the instance  $\mathcal{I}'$  as described in Section 3.2. Then, we run the assignment algorithm by [8] on  $\mathcal{I}'$ . Let  $\mathcal{I}''$  be the resulting instance. By Theorem 4.1,  $\mathcal{I}''$  is 3-approximately fair. By Corollary 4.8, we can pick a subset  $S$  of data points such that  $|S| = O(k^2f^2)$  and there exists an solution  $r'$  of cost at most  $2\text{OPT}_{\mathcal{I}''}$  that only moves points in  $S$ . Now, we apply the dynamic programming algorithm in Theorem 1.3 on  $S$  to obtain a solution of cost at most  $O(\log k)\text{OPT}_{\mathcal{I}''}$  with high probability that runs in  $|S|^{O(\ell)} \log n = (kf)^{O(\ell)} \log n$  time. Since all the previous steps take  $\text{poly}(n)$  time, the total running time of this algorithm is  $\text{poly}(n) + (kf)^{O(\ell)} \log n$ .

Now, by Lemma 4.2 and Claim 3.2, the total cost of our solution is  $O(\log k)\text{OPT}_{\mathcal{I}}$ .  $\square$

## 5 CONCLUSIONS

In this paper, we study the fair  $k$ -median problem. We present an  $O(\log k)$ -approximation algorithm that runs in time  $n^{O(\ell)}$ . We further showed that our algorithm works in a more general setting where the fairness requirements are specified as an arbitrary set of fair profiles. This notion “profile-based fairness” captures a

richer class of fairness requirements that cannot be handled by the previously known approaches for fair representation clustering.

In addition, in the special case of exact fairness, we present an  $O(\log k)$ -approximation algorithm that runs in  $(kf)^{O(\ell)} \log n + \text{poly}(n)$  time, where  $f$  is the size of a fairlet.

Our paper shows that there exists approximation algorithms for fair representation clustering with  $O(1)$  protected classes that run in polynomial time. It remains as an exciting question whether polynomial time  $O(\log k)$ -approximation can be achieved for the problem when  $\ell = \Omega(1)$ . An approach that can be potentially helpful to resolve the previous question is to extend the “reassignment” method used in Section 4 to the more general setting of representation fairness (as in Section 3). Another interesting direction is to extend our results to  $k$ -means problem. A natural approach is to consider the semi-metric  $d'(x, y) = d^2(x, y)$  and apply our DP algorithm to  $d'$ . The problem of this approach is that when we embed the metric to a distribution of tree metrics, we need to bound  $Ed_{\chi}^2$  by  $(Ed_{\chi})^2$ , which is not true.

## ACKNOWLEDGMENTS

Zhen Dai was funded by DARPA (grant number: HR00112190040) and NSF (grant DMS 1854831).

## REFERENCES

- [1] Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. 2021. Fair Clustering via Equitable Group Representations. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 504–514.
- [2] Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. 2019. Clustering without over-representation. In *Proceedings of the SIGKDD International Conference on Knowledge Discovery & Data Mining*. 267–275.
- [3] Haris Angelidakis, Konstantin Makarychev, and Yuri Makarychev. 2017. Algorithms for stable and perturbation-resilient problems. In *Proceedings of the Symposium on Theory of Computing*. 438–451.
- [4] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. 2019. Scalable Fair Clustering. In *Proceedings of the International Conference on Machine Learning*. 405–413.
- [5] Sayan Bandyopadhyay, Fedor V. Fomin, and Kirill Simonov. 2021. On Coresets for Fair Clustering in Metric and Euclidean Spaces and Their Applications. In *Proceedings of the International Colloquium on Automata, Languages, and Programming*. 23:1–23:15.
- [6] Yair Bartal. 1996. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the Conference on Foundations of Computer Science*. 184–193.
- [7] Yair Bartal. 1998. On approximating arbitrary metrics by tree metrics. In *Proceedings of the Symposium on Theory of Computing*. 161–168.
- [8] Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. 2019. Fair algorithms for clustering. In *Advances in Neural Information Processing Systems*. 4955–4966.
- [9] Ioana O Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rönsner, Daniel R Schmidt, and Melanie Schmidt. 2019. On the Cost of Essentially Fair Clusterings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*.
- [10] Matteo Böhm, Adriano Fazzzone, Stefano Leonardi, and Chris Schwegelshohn. 2020. Fair clustering with multiple colors. *arXiv preprint arXiv:2002.07892* (2020).
- [11] Brian Brubach, Darshan Chakrabarti, John Dickerson, Samir Khuller, Aravind Srinivasan, and Leonidas Tsepeneas. 2020. A Pairwise Fair and Community-preserving Approach to  $k$ -Center Clustering. In *Proceedings of the International Conference on Machine Learning*. 1178–1189.
- [12] Deeparnab Chakrabarty and Maryam Negahbani. 2021. Better Algorithms for Individually Fair  $k$ -Clustering. *arXiv preprint arXiv:2106.12150* (2021).
- [13] Moses Charikar, Sudipto Guha, Éva Tardos, and David B Shmoys. 2002. A constant-factor approximation algorithm for the  $k$ -median problem. *J. Comput. System Sci.* 65, 1 (2002), 129–149.
- [14] Xingyu Chen, Brandon Fain, Liang Lyu, and Kamesh Munagala. 2019. Proportionally fair clustering. In *International Conference on Machine Learning*. PMLR, 1032–1041.
- [15] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2017. Fair Clustering Through Fairlets. In *Advances in Neural Information Processing*

- Systems*. 5036–5044.
- [16] Eden Chlamtáč, Yury Makarychev, and Ali Vakilian. 2022. Approximating Fair Clustering with Cascaded Norm Objectives. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2664–2683.
- [17] Alexandra Chouldechova. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data* 5, 2 (2017), 153–163.
- [18] Alexandra Chouldechova and Aaron Roth. 2020. A snapshot of the frontiers of fairness in machine learning. *Commun. ACM* 63, 5 (2020), 82–89.
- [19] Seyed Esmaili, Brian Brubach, Leonidas Tsepenekas, and John Dickerson. 2020. Probabilistic fair clustering. *Advances in Neural Information Processing Systems* 33 (2020).
- [20] Seyed A Esmaili, Brian Brubach, Aravind Srinivasan, and John P Dickerson. 2021. Fair Clustering Under a Bounded Cost. *Advances in Neural Information Processing Systems* (2021).
- [21] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. 2004. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. System Sci.* 69, 3 (2004), 485–497.
- [22] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining*. 259–268.
- [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] Lingxiao Huang, Shaofeng Jiang, and Nisheeth Vishnoi. 2019. Coresets for Clustering with Fairness Constraints. In *Proceedings of the Conference on Neural Information Processing Systems*.
- [25] Matthew Jones, Huy Nguyen, and Thy Nguyen. 2020. Fair  $k$ -centers via maximum matching. In *Proceedings of the International Conference on Machine Learning*. 4940–4949.
- [26] Christopher Jung, Sampath Kannan, and Neil Lutz. 2020. A Center in Your Neighborhood: Fairness in Facility Location. In *Proceedings of the Symposium on Foundations of Responsible Computing*. 5:1–5:15.
- [27] O. Kariv and S. L. Hakimi. 1979. An Algorithmic Approach to Network Location Problems. II: The  $p$ -Medians. *SIAM J. Appl. Math.* 37, 3 (1979), 539–560.
- [28] Michael Kearns and Aaron Roth. 2019. *The ethical algorithm: The science of socially aware algorithm design*. Oxford University Press.
- [29] Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. 2018. Human decisions and machine predictions. *The quarterly journal of economics* 133, 1 (2018), 237–293.
- [30] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *Proceedings of the Innovations in Theoretical Computer Science*.
- [31] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. 2019. Fair  $k$ -Center Clustering for Data Summarization. In *Proceedings of the International Conference on Machine Learning*. 3448–3457.
- [32] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. 2020. A Notion of Individual Fairness for Clustering. *arXiv preprint arXiv:2006.04960* (2020).
- [33] Shi Li and Ola Svensson. 2016. Approximating  $k$ -median via pseudo-approximation. *SIAM J. Comput.* 45, 2 (2016), 530–547.
- [34] Sepideh Mahabadi and Ali Vakilian. 2020. Individual fairness for  $k$ -clustering. In *Proceedings of the International Conference on Machine Learning*. 6586–6596.
- [35] Yury Makarychev and Ali Vakilian. 2021. Approximation Algorithms for Socially Fair Clustering. In *Proceedings of the Conference on Learning Theory*. PMLR, 3246–3264.
- [36] Frank Marcinkowski, Kimon Kieslich, Christopher Starke, and Marco Lünich. 2020. Implications of AI (un-) fairness in higher education admissions: the effects of perceived AI (un-) fairness on exit, voice and organizational reputation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 122–130.
- [37] Evi Micha and Nisarg Shah. 2020. Proportionally Fair Clustering Revisited. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- [38] Clemens Rösner and Melanie Schmidt. 2018. Privacy Preserving Clustering with Constraints. In *Proceedings of the International Colloquium on Automata, Languages, and Programming*. 96:1–96:14.
- [39] Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. 2019. Fair Coresets and Streaming Algorithms for Fair  $k$ -means. In *Proceedings of the International Workshop on Approximation and Online Algorithms*. 232–251.
- [40] Arie Tamir. 1996. An  $O(pn^2)$  algorithm for the  $p$ -median and related problems on tree graphs. *Operations Research Letters* 19, 2 (1996), 59–64.
- [41] Ali Vakilian and Mustafa Yalçın. 2021. Improved approximation algorithms for individually fair clustering. *arXiv preprint arXiv:2106.14043* (2021).