Diversity-aware k-median: Clustering with fair center representation*

Suhas Thejaswi[1], Bruno Ordozgoiti[1], and Aristides Gionis[1,2]

[1] Aalto University, Espoo, Finland.
su.thejaswi@gmail.com, bruno.ordozgoiti@aalto.fi
[2] KTH Royal Institute of Technology, Stockholm, Sweden.
argioni@kth.se

Abstract. We introduce a novel problem for diversity-aware clustering. We assume that the potential cluster centers belong to a set of groups defined by protected attributes, such as ethnicity, gender, etc. We then ask to find a minimum-cost clustering of the data into k clusters so that a specified minimum number of cluster centers are chosen from each group. We thus require that all groups are represented in the clustering solution as cluster centers, according to specified requirements. More precisely, we are given a set of clients C, a set of facilities $\mathcal{F}$, a collection $\mathscr{F} = \{F_1, \ldots, F_t\}$ of facility groups $F_i \subseteq \mathcal{F}$, a budget k, and a set of lower-bound thresholds $R = \{r_1, \ldots, r_t\}$, one for each group in $\mathscr{F}$. The *diversity-aware k-median problem* asks to find a set S of k facilities in $\mathcal{F}$ such that $|S \cap F_i| \geq r_i$, that is, at least r_i centers in S are from group F_i, and the k-median cost $\sum_{c \in C} \min_{s \in S} d(c, s)$ is minimized. We show that in the general case where the facility groups may overlap, the diversity-aware k-median problem is **NP**-hard, fixed-parameter intractable with respect to parameter k, and inapproximable to any multiplicative factor. On the other hand, when the facility groups are disjoint, approximation algorithms can be obtained by reduction to the *matroid median* and *red-blue median* problems. Experimentally, we evaluate our approximation methods for the tractable cases, and present a relaxation-based heuristic for the theoretically intractable case, which can provide high-quality and efficient solutions for real-world datasets.

Keywords: Algorithmic bias · Algorithmic fairness · Diversity-aware clustering · Fair clustering.

1 Introduction

As many important decisions are being automated, algorithmic fairness is becoming increasingly important. Examples of critical decision-making systems

* This research is supported by the Academy of Finland projects AIDA (317085) and MLDB (325117), the ERC Advanced Grant REBOUND (834862), the EC H2020 RIA project SoBigData (871042), and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

include determining credit score for a consumer, computing risk factors for an insurance, pre-screening applicants for a job opening, dispatching patrols for predictive policing, and more. When using algorithms to make decisions for such critical tasks, it is essential to design and employ methods that minimize bias and avoid discrimination against people based on gender, race, or ethnicity.

Algorithmic fairness has gained wide-spread attention in the recent years [20]. The topic has been predominantly studied for *supervised machine learning*, while fairness-aware formulations have also been proposed for *unsupervised machine learning*, for example, *fair clustering* [3, 5, 9, 23], *fair principal component analysis* [22], or *fair densest-subgraph mining* [1]. For the clustering problem the most common approach is to incorporate fairness by the means of *representation-based constraints*, i.e., requiring that all clusters contain certain proportions of the different groups in the data, where data groups are defined via a set of protected attributes, such as demographics. In this paper we introduce a novel notion for fair clustering based on *diversity constraints on the set of selected cluster centers*.

Research has revealed that bias can be introduced in machine-learning algorithms when bias is present in the input data used for training, and methods are designed without considerations for diversity or constraints to enforce fairness [11]. A natural solution is to introduce diversity constraints. We can look at diversification from two different perspectives: (*i*) avoiding over-representation; and (*ii*) avoiding under-representation. In this paper we focus on the latter requirement. Even though these two approaches look similar, a key contribution in our work is to observe that they are mathematically distinct and lead to computational problems having different complexity; for details see Sections 3 and 4.

To motivate our work we present two application scenarios.

Committee selection: We often select committees to represent an underlying population and work towards a task, e.g., a program committee to select papers for a conference, or a parliamentary committee to handle an issue. As we may require that each member of the population is represented by at least one committee member, it is natural to formalize the committee-selection task as a clustering problem, where the committee members will be determined by the centers of the clustering solution. In addition, one may require that the committee is composed by a diverse mix of the population with no under-represented groups, e.g., a minimum fraction of the conference PC members work in industry, or a minimum fraction of the parliamentary committee are women.

News-articles summarization: Consider the problem of summarizing a collection of news articles obtained, for example, as a result to a user query. Clustering these articles using a bag-of-words representation will allow us to select a subset of news articles that cover the different topics present in the collection. In addition, one may like to ensure that the representative articles comes from a diverse set of media sources, e.g., a minimum fraction of the articles comes from left-leaning media or from opinion columns.

To address the scenarios discussed in the previous two examples, we introduce a novel formulation of diversity-aware clustering with representation constraints

Table 1: An overview of our results. All problem cases we consider are **NP**-hard. FPT(k) indicates whether the problem is fixed-parameter tractable with respect to parameter k. *Approx. factor* shows the factor of approximation obtained, and *Approx. method* shows the method used.

Problem	**NP**-hard	FPT(k)	Approx. factor	Approx. method
	Intractable case: intersecting facility groups			
General variant	✓	✗	inapproximable	
	Tractable cases: disjoint facility groups			
$t > 2$, $\sum_{i\in[t]} r_i = k$	✓	open	8	LP
$t > 2$, $\sum_{i\in[t]} r_i < k$	✓	open	8	$\mathcal{O}(k^{t-1})$ calls to LP
$t = 2$, $r_1 + r_2 = k$	✓	open	$3+\epsilon$	local search
$t = 2$, $r_1 + r_2 < k$	✓	open	$3+\epsilon$	$\mathcal{O}(k)$ calls to local search

on cluster centers. In particular, we assume that a set of groups is associated with the facilities to be used as cluster centers. Facility groups may correspond to demographic groups, in the first example, or to types of media sources, in the second. We then ask to cluster the data by selecting a subset of facilities as cluster centers, such that the clustering cost is minimized, and requiring that each facility group is *not under-represented* in the solution.

We show that in the general case, where the facility groups overlap, the diversity-aware k-median problem is not only **NP**-hard, but also fixed-parameter intractable with respect to the number of cluster centers, and inapproximable to any multiplicative factor. In fact, we prove it is **NP**-hard to even find a feasible solution, that is, a set of centers which satisfies the representation constraints, regardless of clustering cost. These hardness results set our clustering problem in stark contrast with other clustering formulations where approximation algorithms exist, and in particular, with the *matroid-median problem* [8, 19, 16], where one asks that facility groups are *not over-represented*. Unfortunately, however, the matroid-median problem does not ensure fairness for all facility groups.

On the positive side, we identify important cases for which the diversity-aware k-median problem is approximable, and we devise efficient algorithms with constant-factor approximation guarantees. These more tractable cases involve settings when the facility groups are disjoint. Even though the general variant of the problem in inapproximable, we demonstrate using experiments that we can obtain a desired clustering solution with representation constraints with almost the same cost as the unconstrained version using simple heuristics based on local-search. The hardness and approximability results for the diversity-aware k-median problem are summarized in Table 1.

In addition to our theoretical analysis and results we empirically evaluate our methods on several real-world datasets. Our experiments show that in many problem instances, both theoretically tractable and intractable, the *price of diversity* is low in practice (See Section 6.1 for a precise definition). In particular, our methods can be used to find solutions over a wide range of diversity requirements where the clustering cost is comparable to the cost of unconstrained clustering.

The rest of this paper is structured as follows. Section 2 discusses related work, Section 3 presents the problem statement and computational complexity results. Section 4 discusses special cases of the problem that admit polynomial-time approximations. In Section 5 we offer efficient heuristics and related tractable objectives, and in Section 6 we describe experimental results. Finally, Section 7 is a short conclusion.

2 Related work

Algorithmic fairness has attracted a considerable amount of attention in recent years, as many decisions that affect us in everyday life are being made by algorithms. Many machine-learning and data-mining problems have been adapted to incorporate notions of fairness. Examples include problems in classification [4, 13, 17], ranking [24, 28], recommendation systems [27], and more.

In this paper we focus on the problem of clustering and we consider a novel notion of fairness based on diverse representation of cluster centers. Our approach is significantly different (and orthogonal) from the standard notion of *fair clustering*, introduced by the pioneering work of Chierichetti et al. [9]. In that setting, data points are partitioned into groups (Chierichetti et al. considered only two groups) and the fairness constraint is that each cluster should contain a fraction of points from each group. Several recent papers have extended the work of Chierichetti et al. by proposing more scalable algorithms [3, 18], extending the methods to accommodate more than two groups [6, 23], or introducing privacy-preserving properties [21]. In this line of work, the fairness notion applies to the representation of data groups within each cluster. In contrast, in our paper the fairness notion applies to the representation of groups in the cluster centers.

The closest-related work to our setting are the problems of *matroid median* [8, 19] and *red-blue median* [16, 15], which can be used to ensure that no data groups are over-represented in the cluster centers of the solution — in contrast we require that no data groups are under-represented. Although the two notions are related, in a way that we make precise in Section 4, in the general case they differ significantly, and they yield problems of greatly different complexity. Furthermore, in the general case, the matroid-median problem cannot be used to ensure fair representation, as upper-bound constraints cannot ensure that all groups will be represented in the solution. Although it is for those cases that our diversity-aware clustering problem is intractable, one can develop practical heuristics that achieve fair results, with respect to diverse representation, as shown in our experimental evaluation.

3 Problem statement and complexity

We consider a set of clients C and a set of facilities $\mathcal{F}$. In some cases, the set of facilities may coincide with the set of clients ($\mathcal{F} = C$), or it is a subset ($\mathcal{F} \subseteq C$). We assume a distance function $d : C \times \mathcal{F} \to \mathbb{R}_+$, which maps client–facility pairs into nonnegative real values. We also consider a collection $\mathscr{F} = \{F_1, \ldots, F_t\}$ of facility groups $F_i \subseteq \mathcal{F}$. During our discussion we distinguish different cases for the structure of $\mathscr{F}$. In the most general case the facility groups F_i may overlap. Two special cases of interest, discussed in Section 4, are when the facility groups F_i are disjoint and when there are only two groups. Finally, we are given a total budget k of facilities to be selected, and a set of lower-bound thresholds $R = \{r_1, \ldots, r_t\}$, i.e., one threshold for each group F_i in $\mathscr{F}$.

The diversity-aware k-median problem (Div-k-Median) asks for a set S of k facilities in $\mathcal{F}$ subject to the constraint $|S \cap F_i| \geq r_i$, such that the k-median cost $\sum_{c \in C} \min_{s \in S} d(c, s)$ is minimized. Thus, we search for a minimum-cost clustering solution S where each group F_i is represented by at least r_i centers.

In the following sections, we study the computational complexity of the Div-k-Median problem. In particular, we show that the general variant of the problem is (*i*) **NP**-hard; (*ii*) not fixed-parameter tractable with respect to parameter k, i.e., the size of the solution sought; and (*iii*) inapproximable to any multiplicative factor. In fact, we show that hardness results (*i*) and (*ii*) apply for the problem of simply finding a feasible solution. That is, in the general case, and assuming $\mathbf{P} \neq \mathbf{NP}$ there is no polynomial-time algorithm to find a solution $S \subseteq \mathcal{F}$ that satisfies the constraints $|S \cap F_i| \geq r_i$, for all $i \in [t]$. The inapproximability statement (*iii*) is a consequence of the **NP**-hardness for finding a feasible solution. These hardness results motivate the heuristics we propose later on.

3.1 NP-hardness

We prove **NP**-hardness by reducing the dominating set problem to the problem of finding a *feasible solution* to Div-k-Median.

Dominating set problem (DomSet). Given a graph $G = (V, E)$ with $|V| = n$ vertices, and an integer $k \leq n$, decide if there exists a subset $S \subseteq V$ of size $|S| = k$ such that for each $v \in V$ it is either $\{v\} \cap S \neq \emptyset$ or $\{v\} \cap N(S) \neq \emptyset$, where $N(S)$ denotes the set of vertices adjacent to at least one vertex in S. In other words, each vertex in V is either in S or adjacent to at least one vertex in S.

Lemma 1. *Finding a feasible solution for* Div-k-Median *is* **NP**-*hard.*

Proof. Given an instance of DomSet $(G = (V, E), k)$, we construct an instance of the Div-k-Median problem $(C, \mathcal{F}, \mathscr{F}, d, k, R)$, such that $C = V$, $\mathcal{F} = V$, $d(u, v) = 1$ for all $(u, v) \in C \times F$, $\mathscr{F} = \{F_1, \ldots, F_n\}$ with $F_u = \{u\} \cup N(u)$, and $R = \{1, \ldots, 1\}$, i.e., the lower-bound thresholds are set to $r_u = 1$, for all $u \in V$.

Let $S \subseteq C$ be a feasible solution for Div-k-Median. From the construction it is clear that S is a dominating set, as $|F_u \cap S| \geq 1$, and thus S intersects $\{u\} \cup N(u)$ for all $u \in V$. The proof that a dominating set is a feasible solution to Div-k-Median is analogous. □

The hardness of diversity-aware k-median follows immediately.

Corollary 1. *The* DIV-k-MEDIAN *problem is* **NP***-hard.*

3.2 Fixed-parameter intractability

A problem P specified by input x and a parameter k is *fixed-parameter tractable* (FPT) if there exists an algorithm A to solve every instance $(x, k) \in P$ with running time of the form $f(k)|x|^{\mathcal{O}(1)}$, where $f(k)$ is function depending solely on the parameter k and $|x|^{\mathcal{O}(1)} = \text{poly}(|x|)$ is a polynomial independent of the parameter k. A problem P is *fixed-parameter intractable* with respect to parameter k otherwise.

To show that the DIV-k-MEDIAN is fixed-parameter intractable we present a *parameterized reduction* from the DOMSET problem to DIV-k-MEDIAN.[3] The DOMSET problem is known to be fixed-parameter intractable [10, Theorem 13.9]. This means that there exists no algorithm with running time $f(k)|V|^{\mathcal{O}(1)}$ to solve DOMSET, where $f(k)$ is a function depending solely on the parameter k.

Theorem 1. *The* DIV-k-MEDIAN *problem is fixed-parameter intractable with respect to the parameter k, that is, the size of the solution sought.*

Proof. We apply the reduction from Lemma 1. It follows that (i) an instance (G, k) of the DOMSET problem has a feasible solution if and only if there exists a feasible solution for the DIV-k-MEDIAN problem instance $(C, \mathcal{F}, \mathscr{F}, d, k', R)$, with $k' = k$, and (ii) the reduction takes polynomial time in the size of the input. So there exists a parameterized reduction from the DOMSET problem to the DIV-k-MEDIAN problem. This implies that if there exists an algorithm with running time $f(k')|C|^{\mathcal{O}(1)}$ for the DIV-k-MEDIAN problem then there exits an algorithm with running time $f(k)|V|^{\mathcal{O}(1)}$ for solving the DOMSET problem. □

It would still be interesting to check whether there exists a parameter of the problem that can be used to design a solution where the exponential complexity can be restricted. We leave this as an open problem.

3.3 Hardness of approximation

We now present hardness-of-approximation results for DIV-k-MEDIAN.

Theorem 2. *Assuming* $\mathbf{P} \neq \mathbf{NP}$*, the* DIV-$k$-MEDIAN *problem cannot be approximated to any multiplicative factor.*

Proof. We apply the reduction of the DOMSET problem from the proof of Lemma 1. For the sake of contradiction let A be a polynomial-time approximation algorithm which gives a factor-c approximate solution for DIV-k-MEDIAN. Then we can employ algorithm A to obtain an exact solution to the DOMSET instance in polynomial time, by way of the aforementioned reduction. The reason is that an approximate solution for DIV-k-MEDIAN is also a feasible solution, which in turn implies a feasible solution for DOMSET. Thus, unless $\mathbf{P} \neq \mathbf{NP}$, DIV-$k$-MEDIAN cannot be approximated to any multiplicative factor. □

[3] For a precise definition of parameterized reduction see Cygan et al. [10, Chapter 13].

4 Approximable instances

In Section 3 we presented strong intractability results for the DIV-k-MEDIAN problem. Recall that inapproximability stems from the fact that satisfying the non under-representation constraints $|S \cap F_i| \geq r_i$ for $i \in [t]$ is **NP**-hard. *So the inapproximability holds even if we change the clustering cost, for instance, to k-center or a soft assignment variant of k-median.*[4] Fortunately, however, there are instances where finding a feasible solution is polynomial-time solvable, even if finding an minimum-cost clustering solution remains **NP**-hard. In this section we discuss such instances and give approximation algorithms.

4.1 Non-intersecting facility groups

We consider instances of DIV-k-MEDIAN where $F_i \cap F_j = \emptyset$ for all $F_i, F_j \in \mathscr{F}$, that is, the facility groups are disjoint. We refer to variants satisfying disjointness conditions as the DIV-k-MEDIAN$_\emptyset$ problem.

A feasible solution exists for DIV-k-MEDIAN$_\emptyset$ if and only if $|F_i| \geq r_i$ for all $i \in [t]$ and $\sum_{i\in[t]} r_i \leq k$. Furthermore, assuming that the two latter conditions hold true, finding a feasible solution is trivial: it can be done simply by picking r_i facilities from each facility group F_i.

It can be shown that the DIV-k-MEDIAN$_\emptyset$ problem can be reduced to the *matroid-median problem* [19], and use existing techniques for the latter problem to obtain an 8-approximation algorithm for DIV-k-MEDIAN$_\emptyset$ [25]. Before discussing the reduction we first introduce the matroid-median problem.

The matroid-median problem (MATROIDMEDIAN) [19]. We are given a finite set of clients C and facilities $\mathcal{F}$, a metric distance function $d : C \times \mathcal{F} \to \mathbb{R}_+$, and a matroid $\mathcal{M} = (\mathcal{F}, \mathcal{I})$ with ground set $\mathcal{F}$ and a collection of independent sets $\mathcal{I} \subseteq 2^{\mathcal{F}}$. The problem asks us to find a subset $S \in \mathcal{I}(\mathcal{M})$ such that the cost function $\text{cost}(S) = \sum_{c\in C} \min_{s\in S} d(c, s)$ is minimized.

The MATROIDMEDIAN problem is a generalization of k-median, and has an 8-approximation algorithm based on LP relaxation [25]. Here we present a reduction of DIV-k-MEDIAN$_\emptyset$ to MATROIDMEDIAN. In this section we handle the case where $\sum_{i\in[t]} r_i = k$. In Section 4.3 we show that the case $\sum_{i\in[t]} r_i < k$ can be reduced to the former one with at most $\mathscr{O}(k^{t-1})$ calls. Approximating DIV-k-MEDIAN$_\emptyset$ in polynomial-time when $\sum_{i\in[t]} r_i < k$ is left open.

The reduction. Given an instance $(C, \mathcal{F}, \mathscr{F}, d, k, R)$, of the DIV-$k$-MEDIAN$_\emptyset$ problem we generate an instance $(C', \mathcal{F}', \mathcal{I}', d')$ of the MATROIDMEDIAN problem as follows: $C' = C$, $\mathcal{F}' = \mathcal{F}$, $d' = d$, and $\mathcal{M} = (\mathcal{F}', \mathcal{I}')$ where $\mathcal{I}' \subseteq 2^{\mathcal{F}'}$ and $A \in \mathcal{I}'$ if $|A \cap F_i| \leq r_i$ for all $r_i \in R$. More precisely, the set of independent sets is comprised of all subsets of $\mathcal{F}'$ that satisfy *non over-representation* constraints. It is easy to verify that $\mathcal{M}$ is a matroid — it is a *partition matroid*. In the event that the algorithm for MATROIDMEDIAN outputs a solution where $|A \cap F_i| < r_i$, for some i, since $\sum_{i\in[t]} r_i = k$, it satisfies all the constraints with equality by

[4] In soft clustering, each client is assigned to all cluster centers with a probability.

completing the solution with facilities of the missing group(s) at no additional connection cost. Since we can ensure that $|A \cap F_i| = r_i$, for all i, it also holds $|A \cap F_i| \geq r_i$, for all i, that is, the Div-k-Median$_\emptyset$ constraints.

Since the MatroidMedian problem has a polynomial-time approximation algorithm, it follows from our inapproximability results (Section 3) that a reduction of the general formulation of Div-k-Median is impossible. We can thus conclude that allowing intersections between facility groups fundamentally changes the combinatorial structure of feasible solutions, interfering with the design of approximation algorithms.

4.2 Two facility groups

The approximation guarantee of the Div-k-Median$_\emptyset$ problem can be further improved if we restrict the number of groups to two.

In particular, we consider instances of the Div-k-Median problem where $F_i \cap F_j = \emptyset$, for all $F_i, F_j \in \mathscr{F}$, and $\mathscr{F} = \{F_1, F_2\}$. For simplicity, the facilities F_1 and F_2 are referred to as *red* and *blue* facilities, respectively.

As before, we can assume that $\sum_{i \in [t]} r_i = r_1 + r_2 \leq k$, otherwise the problem has no feasible solution. We first present a local-search algorithm for the case $r_1 + r_2 = k$. In Section 4.3 we show that the case with $r_1 + r_2 < k$ can be reduced to the former one with a linear number of calls for different values of r_1 and r_2. Before continuing with the algorithm we first define the rb-Median problem.

The red-blue median problem (rb-Median). We are given a set of clients C, two disjoint facility sets F_1 and F_2 (referred to as red and blue facilities, respectively), two integers r_1, r_2 and a metric distance function $d : C \times \{F_1 \cup F_2\} \to \mathbb{R}_+$. The problem asks to find a subset $S \subseteq F_1 \cup F_2$ such that $|F_1 \cap S| \leq r_1$, $|F_2 \cap S| \leq r_2$ and the cost function $\text{cost}(S) = \sum_{c \in C} \min_{s \in S} d(c, s)$ is minimized.

The rb-Median problem accepts a $3 + \epsilon$ approximation algorithm based on local-search [15]. The algorithm works by swapping a red-blue pair (r, b) with a red-blue pair (r', b') as long as the cost improves. Note that $(r' = r, b' \neq b)$, $(r' \neq r, b' = b)$ and $(r' \neq r, b' \neq b)$ are valid swap pairs. The reduction of Div-k-Median to rb-Median is similar to the one given above for MatroidMedian. Thus, when the input consists of two non-intersecting facility groups we can obtain a $3 + \epsilon$ approximation of the optimum in polynomial time which follows from the local-search approximation of the rb-Median problem [15].

4.3 The case $\sum_i r_i < k$

The reduction of Div-k-Median$_\emptyset$ to MatroidMedian relies on picking exactly $\sum_i r_i$ facilities. This is because it is not possible to define a matroid that simultaneously enforces the desired lower-bound facility group constraints and the cardinality constraint for the solution. Nevertheless, we can overcome this obstacle at a bounded cost in running time.

1. Initialize S to be an arbitrary feasible solution.
2. While there exists a pair (s, s'), with $s \in S$ and $s' \in \mathcal{F}$ such that
   (a) $\text{cost}(S \setminus \{s\} \cup \{s'\}) < \text{cost}(S)$ and
   (b) $S \setminus \{s\} \cup \{s'\}$ is feasible i.e, $|S \setminus \{s\} \cup \{s'\} \cap F_i| \geq r_i$ for all $i \in [t]$,
   Set $S = S \setminus \{s\} \cup \{s'\}$.
3. Return S.

Fig. 1: Local search heuristic (LS-1) for Div-k-Median$_\emptyset$.

So, in the case that $\sum_i r_i < k$, in order to satisfy the constraint $|S| = k$, we can simply increase the lower-bound group constraints $r_i \mapsto r_i' > r_i$, $i = 1, \ldots, t$ so that $\sum_i r_i' = k$. However, if we do this in an arbitrary fashion we might make a suboptimal choice. To circumvent this, we can exhaustively inspect all possible choices. For this, it suffices to construct $\binom{k - \sum_i r_i + t - 1}{t-1} = \mathcal{O}(k^{t-1})$ instances of MatroidMedian. In the case of rb-Median discussed in Section 4.2, i.e., when $r_1 + r_2 < k$, the required number of instances is linear in k.

5 Proposed methods

In this section we present practical methods to solve the diversity-aware clustering problem. In particular, we present local-search algorithms for Div-k-Median$_\emptyset$ and a method based on relaxing the representation constraints for Div-k-Median.

5.1 Local search

Algorithms based on the local-search heuristic have been used to design approximation algorithms for many optimization problems, including facility location [2, 7, 14] and k-median [2, 7, 15] problems. In light of the inapproximability results presented in the previous section it comes as no surprise that any polynomial-time algorithm, including local-search methods, cannot be expected to find a feasible solution for the Div-k-Median problem. Nevertheless, local-search methods are viable for the tractable instances discussed in Section 4, and can be shown to provide provable quality guarantees.

For solving the Div-k-Median$_\emptyset$ problem we propose two algorithms based on local search.

Local search variant #1 (LS-1). We propose a single-swap local-search algorithm described in Figure 1. The key difference with respect to vanilla local search is that we must ensure that a swap does not violate the representation constraints.

We stress that the proposed algorithm LS-1 is not viable for general instances of Div-k-Median with intersecting facility groups. To illustrate, we present an example in Figure 2. Let F_r, F_g, F_b, F_y be facility groups, corresponding to the colors red, green, blue and yellow, respectively. The intersection cardinality constraints $r_r = r_g = r_b = r_y = 1$ and the number of medians $k = 2$.

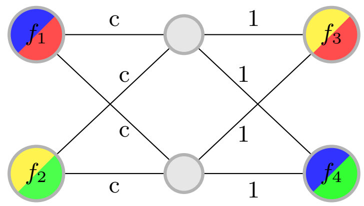

Fig. 2: An example illustrating the infeasibility of local search.

1. Initialize — arbitrarily pick:
   (a) $S_i \subseteq F_i$ such that $|S_i| = r_i$ for all $i \in [t]$,
   (b) $S_{t+1} \subseteq \mathcal{F} \setminus \bigcup_{i\in[t]} S_i$ such that $|S_{t+1}| = k - \sum_{i\in[t]} r_i$, and
   (c) initial solution is $S = \bigcup_{i\in[t]} S_i \cup S_{t+1}$.
2. Iterate — while there exists tuples $(s_1, \ldots, s_{t+1})$ and $(s'_1, \ldots, s'_{t+1})$ such that:
   (a) $s_i \in S_i$, $s'_i \in F_i$ for all $i \in [t]$, $s_{t+1} \in S_{t+1}$, $s'_{t+1} \in \mathcal{F} \setminus \bigcup_{i\in[t]} S_i$
   (b) $S \setminus \{s_1, \ldots, s_{t+1}\} \cup \{s'_1, \ldots, s'_{t+1}\}$ is feasible, and
   (c) $\text{cost}(S \setminus \{s_1, \ldots, s_{t+1}\} \cup \{s'_1, \ldots, s'_{t+1}\}) < \text{cost}(S)$
   set $S = S \setminus \{s_1, \ldots, s_{t+1}\} \cup \{s'_1, \ldots, s'_{t+1}\}$.
3. Return S.

Fig. 3: Local-search heuristic (LS-2) for Div-k-Median$_\emptyset$

Let $S = \{f_1, f_2\}$ be a feasible solution. It is trivial to see that we cannot swap f_1 with either f_3 or f_4, since both swaps violate the constraints $|S \cap F_b| \geq 1$ and $|S \cap F_r| \geq 1$, respectively. Likewise we cannot swap f_2 with either f_3 or f_4. So our local-search algorithm is stuck at a local optima and the approximation ratio is c, which can be made arbitrarily large. We can construct a family of infinitely many such problem instances where the local-search algorithm returns arbitrarily bad results. Similarly we can construct a family of infinitely many instances where the Div-k-Median problem with t facility groups and $k < t$ would require at least $t-1$ parallel swaps to ensure that local search is not stuck in a local optima. This example illustrates the limited applicability of the local-search heuristic for the most general variant of the Div-k-Median problem, where the facility groups overlap in an arbitrary way.

Local search variant #2 (LS-2). Our second approach is the multi-swap local-search heuristic described in Figure 3. The algorithm works by picking r_i facilities from F_i and $k - \sum_{i\in[t]} r_i$ from $\mathcal{F}$ as an initial feasible solution. We swap a tuple of facilities $(s_1, \ldots, s_{t+1})$ with $(s'_1, \ldots, s'_{t+1})$ as long as the cost improves. The algorithm has running time of $\mathcal{O}(n^t)$, and thus it is not practical for large values of t. The algorithm LS-2 is a $3+\epsilon$ approximation for the Div-k-Median$_\emptyset$ problem with two facility groups i.e., $t = 2$ (see Section 4.2). Bounding the approximation ratio of algorithm LS-2 for $t > 2$ is an open problem.

Note that the cost of the solution obtained by LS-1 and LS-2 is the k-median cost, that is, $\text{cost}(S) = \sum_{v\in C} \min_{s\in S} d(v, s)$.

5.2 Relaxing the representation constraints

In view of the difficulty of solving the problem as formulated in Section 3, we explore alternative, more easily optimized formulations to encode the desired representation constraints. We first observe that a straightforward approach, akin to a Lagrangian relaxation, might result in undesirable outcomes. Consider the following objective function:

$$\mathrm{cost}(S) = \sum_{v \in C} \min_{s \in S} d(v, s) + \lambda \sum_{i \in [t]} \max\{r_i - |F_i \cap S|, 0\}, \quad (1)$$

that is, instead of enforcing the constraints, we penalize their violations. A problem with this formulation is that every constraint satisfaction — up to r_i — counts the same, and thus the composition of the solution might be imbalanced.

We illustrate this shortcoming with an example. Consider $\mathscr{F} = \{F_1, F_2, F_3\}$, $k = 6$, $r_1 = 2$, $r_2 = 2$, $r_3 = 0$. Now consider two solutions: (i) 2 facilities from F_1, 0 from F_2, and 4 from F_3; and (ii) 1 facility from F_1, 1 from F_2, and 4 from F_3. Both solutions score the same in terms of number of violations. Nevertheless, the second one is more balanced in terms of group representation. To overcome this issue, we propose the following alternative formulation.

$$\mathrm{cost}_f(S) = \sum_{v \in C} \min_{s \in S} d(v, s) + \lambda \sum_{i \in [t]} \frac{r_i}{|S \cap F_i| + 1}. \quad (2)$$

The second term that encodes the violations enjoys group-level diminishing returns. Thus, when a facility of a protected group is added, facilities from other groups will be favored. The cardinality requirements r_i act here as weights on the different groups.

We optimize the objective in Equation 2 using vanilla local-search by picking an arbitrary initial solution with no restrictions.

6 Experimental evaluation

In order to gain insight on the proposed problem and to evaluate our algorithms, we carried out experiments on a variety of publicly available datasets. Our objective is to evaluate the following key aspects:
Price of diversity: What is the price of enforcing representation constraints? We measure how the clustering cost increases as more stringent requirements on group representation are imposed.
Relaxed objective: We evaluate the relaxation-based method, described in Section 5, for the intractable case with facility group intersections. We evaluate its performance in terms of constraint satisfaction and clustering cost.
Running time: Our problem formulation requires modified versions of standard local-search heuristics, as described in Section 5. We evaluate the impact of these variants on running time.

Table 2: Dataset statistics. n is the number of data points, D is dataset dimension, t is number of facility types. Columns $4, 5$ and $6, 7$ is the maximum and minimum size of facility groups when divided into two disjoint groups and four intersecting groups, respectively.

			$t = 2$		$t = 4$	
Dataset	n	D	min	max	min	max
heart-switzerland	123	14	10	113	-	-
heart-va	200	14	6	194	-	-
heart-hungarian	294	14	81	213	-	-
heart-cleveland	303	14	97	206	-	-
student-mat	395	33	208	187	-	-
house-votes-84	435	17	267	168	-	-
student-por	649	33	383	266	-	-
student-per2	666	12	311	355	-	-
autism	704	21	337	367	20	337
hcv-egy-data	1 385	29	678	707	40	698
cmc	1 473	10	220	1 253	96	511
abalone	4 177	9	1 307	1 342	-	-
mushroom	8 123	23	3 375	4 748	1 072	3 655
nursery	12 959	9	6 479	6 480	3 239	4 320

Datasets. We use datasets from the UCI machine learning repository [12]. We normalize columns to unit norm and use the L_1 metric as distance function. The dataset statistics are reported in Table 2.

Baseline. As a baseline we use a local-search algorithm with no cardinality constraints. We call this baseline `LS-0`. For each dataset we perform 10 executions of `LS-0` with random initial assignments to obtain the solution with minimum cost ℓ_0 among the independent executions. `LS-0` is known to provide a 5-approximation for the k-median problem [2]. We also experimented with *exhaustive enumeration* and *linear program solvers*, however these approaches failed to solve instances with modest size, which is expected given the inherent complexity of Div-k-Median.

Experimental setup. The experiments are executed on a desktop with 4-core *Haswell* CPU and 16 GB main memory. Our source code is written in `Python` and we make use of `numpy` to enable parallelization of computations. Our source code is available as open source [26].

6.1 Results

Price of diversity. For each dataset we identify a protected attribute and classify data points into two disjoint groups. In most datasets we choose gender, except in `house-votes` dataset where we use party affiliation. We identify the smallest group in the dataset (*minority group*) and measure the fraction of the chosen facilities that belong to that group (*minority fraction*). When running

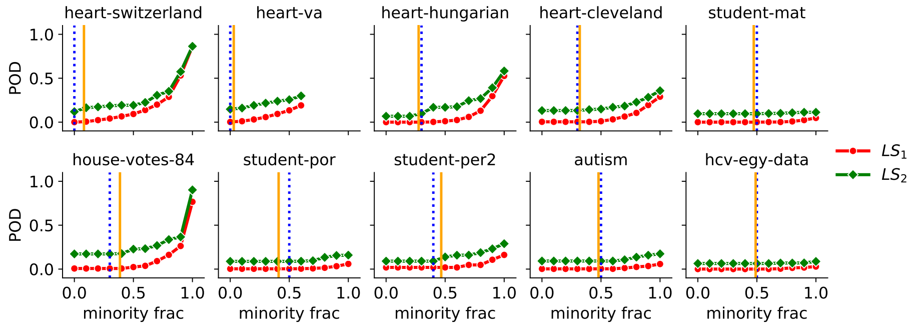

Fig. 4: Price of diversity ($k = 10$).

LS-1 and LS-2 we enforce a specific minority fraction and repeat the experiments for ten iterations by choosing random initial assignments. We refer to the cost of the solutions obtained from LS-0, LS-1 and LS-2 as ℓ_0, ℓ_1 and ℓ_2, respectively.

The *price of diversity* (POD) is the ratio of increase in the cost of the solution to the cost of LS_0 i.e., $\text{POD}(\text{LS-1}) = \frac{\ell_1 - \ell_0}{\ell_0}$ and $\text{POD}(\text{LS-2}) = \frac{\ell_2 - \ell_0}{\ell_0}$. Recall that in theory the POD is unbounded. However, this need not be the case in practical scenarios. Additionally, we compute the differences in group representation between algorithms as follows. Let $R^i = \{r_1^i, \ldots, r_t^i\}$ be the set representing the number of facilities chosen from each group in $\mathscr{F}$ by algorithm LS-j. For $j = 1, 2$ we define $L_1(\text{LS-}j) = \sum_{i \in [t]} |r_i^j - r_i^0| / (kt)$.

In Figure 4, we report the price of diversity (POD) as a function of the imposed minority fraction for LS-1 and LS-2. The blue and yellow vertical lines denote the minority fraction achieved by the baseline LS-0 and the fraction of minority facilities in the dataset, respectively. Notice that the minority fraction of the baseline is very close to the minority fraction of the dataset. With respect to our methods LS-1 and LS-2, we observe little variance among the independent executions. Most importantly, we observe that the price of diversity is relatively low, that is, for most datasets we can vary the representation requirements over a wide range and the clustering cost increases very little compared to the non-constrained version. An increase is observed only for a few datasets and only for extreme values of representation constraints. We also observe that LS-1 outperforms consistently LS-2. This is good news as LS-1 is also more efficient.

In Figure 5, we report the L_1 measure as a function of the increase in the minority fraction. Note that we enforce a restriction that the ratio of minority nodes should be at least the minority fraction, however, the ratio of facilities chosen from the minority group can be more than the minority fraction enforced. In this experiment we measure the change in the type of facilities chosen. We observe more variance in L_1 score among the independent runs when the minority fraction of the solution is less than the minority fraction of the dataset. This shows that the algorithm has more flexibility to choose the different type of facilities. In Figure 7 we report POD and L_1 measure for moderate size datasets.

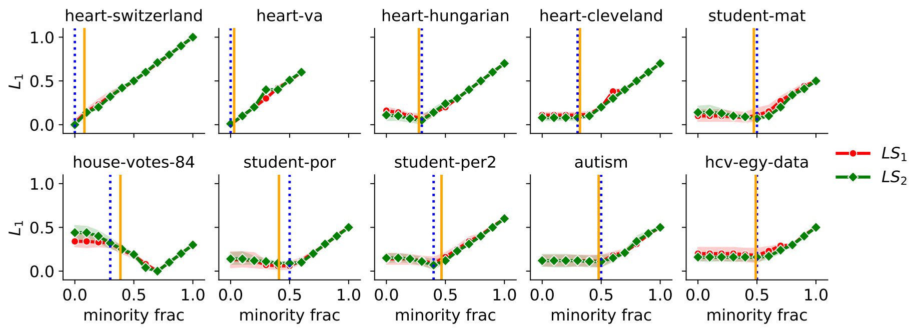

Fig. 5: L_1 distance of the chosen facility types ($k = 10$).

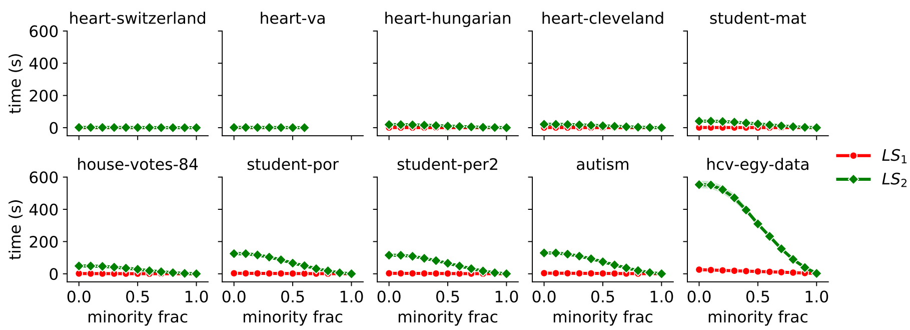

Fig. 6: Running Time ($k = 10$).

Running time. In Figure 6 we report the running time of LS-1 and LS-2 as a function of the minority fraction. For small datasets we observe no significant change in the running time of LS-1 and LS-2. However, the dataset size has a significant impact on running time of LS-2. For instance in the `hcv-egy-data` dataset, for $k = 10$ and minority fraction 0.1, LS-2 is 300 times slower than LS-1. This is expected, as the algorithm considers a quadratic number of replacements per iteration. Despite this increase in time, there is no significant improvement in the cost of the solution obtained, as observed in Figure 4. This makes LS-1 our method of choice in problem instances where the facility groups are disjoint.

Relaxed objective. In our final set of experiments we study the behavior of the LS-0 local-search heuristic with the relaxed objective function of Equation (2). In Figure 8 we report price of diversity (POD) and the fraction of violations of representation constraints L^* for each value of $\lambda = \{2^1, \ldots, 2^7\}$. For each dataset we choose four protected attributes to obtain intersecting facility types, and perform experiments with $k = 10$ and the representation constraints set $R = \{3, 3, 3, 3\}$. The value of L^* measures the fraction of violations of the representation constraints i.e, $L^* = \frac{\sum_{i \in [t]} \min(0, |S \cap F_i| - r_i))}{\sum_{i \in [t]} r_i}$. With the increase in the

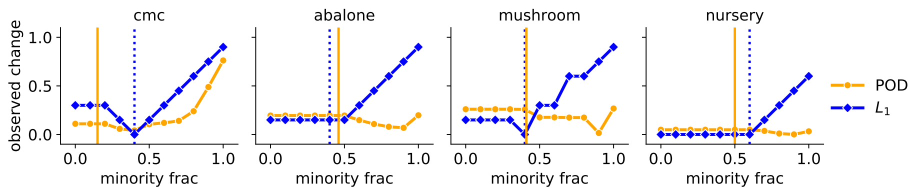

Fig. 7: Price of diversity for moderate size datasets ($k = 10$).

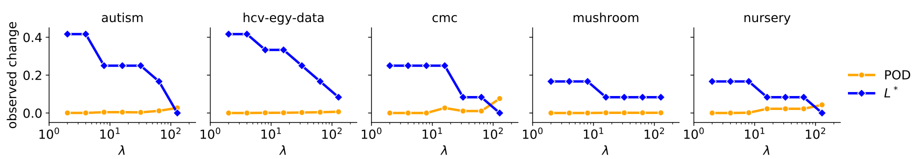

Fig. 8: Price of diversity for intersecting facility types ($k = 10$).

value of λ the value of L^* decreases and the value of POD increases, as expected. However, the increase in POD is very small and in all cases it is possible to find solutions where both POD and L^* are very close to zero, that is, solutions that have very few constraint violations and their clustering cost is almost as low as in the unconstrained version.

7 Conclusion

We introduce a novel formulation of diversity-aware clustering, which ensures fairness by avoiding under-representation of the cluster centers, where the cluster centers belong in different facility groups. We show that the general variant of the problem where facility groups overlap is **NP**-hard, fixed-parameter intractable with respect to the number of clusters, and inapproximable to any multiplicative factor. Despite such negative results we show that the variant of the problem with disjoint facility types can be approximated efficiently. We also present heuristic algorithms that practically solve real-world problem instances and empirically evaluated the proposed solutions using an extensive set of experiments. The main open problem left is to improve the run-time complexity of the approximation algorithm, in the setting of disjoint groups and $t > 2$, so that it does not use repeated calls to a linear program. Additionally, it would be interesting to devise FPT algorithms for obtaining exact solutions, again in the case of disjoint groups.

References

1. Anagnostopoulos, A., Becchetti, L., Fazzone, A., Menghini, C., Schwiegelshohn, C.: Spectral relaxations and fair densest subgraphs. In: CIKM (2020)
2. Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., Pandit, V.: Local-search heuristic for k-median and facility-location problems. In: STOC (2001)

3. Backurs, A., Indyk, P., Onak, K., Schieber, B., Vakilian, A., Wagner, T.: Scalable fair clustering. In: ICML (2019)
4. Barocas, S., Hardt, M., Narayanan, A.: Fairness and Machine Learning. fairmlbook.org (2019)
5. Bera, S., Chakrabarty, D., Flores, N., Negahbani, M.: Fair algorithms for clustering. In: NeurIPS (2019)
6. Bercea, I.O., Groß, M., Khuller, S., Kumar, A., Rösner, C., Schmidt, D.R., Schmidt, M.: On the cost of essentially fair clusterings. In: APPROX (2019)
7. Charikar, M., Guha, S.: Improved combinatorial algorithms for the facility location and k-median problems. In: FOCS (1999)
8. Chen, D.Z., Li, J., Liang, H., Wang, H.: Matroid and knapsack center problems. Algorithmica **75**(1), 27–52 (2016)
9. Chierichetti, F., Kumar, R., Lattanzi, S., Vassilvitskii, S.: Fair clustering through fairlets. In: NeurIPS (2017)
10. Cygan, M., Fomin, F.V., Kowalik, Ł., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized algorithms, vol. 4. Springer (2015)
11. Danks, D., London, A.J.: Algorithmic bias in autonomous systems. In: IJCAI (2017)
12. Dua, D., Graff, C.: UCI machine learning repository (2017), `http://archive.ics.uci.edu/ml`
13. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: ITCS (2012)
14. Gupta, A., Tangwongsan, K.: Simpler analyses of local-search algorithms for facility location. arXiv:0809.2554 (2008)
15. Hajiaghayi, M., Khandekar, R., Kortsarz, G.: Local-search algorithms for the red-blue median problem. Algorithmica **63**(4), 795–814 (2012)
16. Hajiaghayi, M., Khandekar, R., Kortsarz, G.: Budgeted red-blue median and its generalizations. In: ESA (2010)
17. Hardt, M., Price, E., Srebro, N.: Equality of opportunity in supervised learning. In: NeurIPS (2016)
18. Huang, L., Jiang, S., Vishnoi, N.: Coresets for clustering with fairness constraints. In: NeurIPS (2019)
19. Krishnaswamy, R., Kumar, A., Nagarajan, V., Sabharwal, Y., Saha, B.: The matroid median problem. In: SODA (2011)
20. Pedreshi, D., Ruggieri, S., Turini, F.: Discrimination-aware data mining. In: KDD (2008)
21. Rösner, C., Schmidt, M.: Privacy preserving clustering with constraints. In: ICALP (2018)
22. Samadi, S., Tantipongpipat, U., Morgenstern, J., Singh, M., Vempala, S.: The price of fair PCA: one extra dimension. In: NeurIPS (2018)
23. Schmidt, M., Schwiegelshohn, C., Sohler, C.: Fair coresets and streaming algorithms for fair k-means. In: WAOA (2019)
24. Singh, A., Joachims, T.: Policy learning for fairness in ranking. In: NeurIPS (2019)
25. Swamy, C.: Improved approximation algorithms for matroid and knapsack median problems and applications. ACM Transactions of Algorithms **12**(4) (2016)
26. Thejaswi, S., Ordozgoiti, B., Gionis, A.: Diversity-aware k-median : experimental v1.0. `https://github.com/suhastheju/diversity-aware-k-median` (2021)
27. Yao, S., Huang, B.: Beyond parity: Fairness objectives for collaborative filtering. In: NeurIPS (2017)
28. Zehlike, M., Bonchi, F., Castillo, C., Hajian, S., Megahed, M., Baeza-Yates, R.: Fa*ir: A fair top-k ranking algorithm. In: CIKM (2017)