Fair Projections as a Means Towards Balanced Recommendations

ARIS ANAGNOSTOPOULOS, Sapienza, University of Rome, Italy
LUCA BECCHETTI, Sapienza, University of Rome, Italy
MATTEO BÖHM, Sapienza, University of Rome, Italy
ADRIANO FAZZONE, CENTAI, Italy
STEFANO LEONARDI, Sapienza, University of Rome, Italy
CRISTINA MENGHINI, Brown University, USA
CHRIS SCHWIEGELSHOHN, Aarhus University, Denmark

The goal of recommender systems is to provide to users suggestions that match their interests, with the eventual goal of increasing their satisfaction, as measured by the number of transactions (clicks, purchases, etc.). Often, this leads to providing recommendations that are of a particular type. For some contexts (e.g., browsing videos for information) this may be undesirable, as it may enforce the creation of filter bubbles. This is because of the existence of underlying bias in the input data of prior user actions.

Reducing hidden bias in the data and ensuring fairness in algorithmic data analysis has recently received significant attention. In this paper, we consider both the densest subgraph and the k-clustering problem, two primitives that are being used by some recommender systems. We are given a coloring on the nodes, respectively the points, and aim to compute a *fair* solution S, consisting of a subgraph or a clustering, such that none of the colors is disparately impacted by the solution.

Unfortunately, introducing fair solutions typically makes these problems substantially more difficult. Unlike the unconstrained densest subgraph problem, which is solvable in polynomial time, the fair densest subgraph problem is NP-hard even to approximate. For k-clustering, the fairness constraints make the problem very similar to capacitated clustering, which is a notoriously hard problem to even approximate.

Despite such negative premises, we are able to provide positive results in important use cases. In particular, we are able to prove that a suitable spectral embedding allows recovery of an almost optimal, fair, dense subgraph hidden in the input data, whenever one is present, a result that is further supported by experimental evidence.

We also show a polynomial-time, 2-approximation algorithm to the problem of fair densest subgraph, assuming that there exist only two colors and both colors occur equally often in the graph. This result turns out to be optimal assuming the small set expansion hypothesis. For fair k-clustering, we show that we can recover high quality fair clusterings effectively and efficiently. For the special case of k-median and k-center, we offer additional, fast and simple approximation algorithms as well as new hardness results.

The above theoretical findings drive the design of heuristics, which we experimentally evaluate on a scenario based on real data, in which our aim is to strike a good balance between diversity and highly correlated items from Amazon co-purchasing graphs and facebook contacts. We additionally evaluated our algorithmic solutions for the fair k-median problem through experiments on various real-world datasets.

Authors' addresses: Aris Anagnostopoulos, Sapienza, University of Rome, Italy, aris@diag.uniroma1.it; Luca Becchetti, Sapienza, University of Rome, Italy, becchetti@diag.uniroma1.it; Matteo Böhm, Sapienza, University of Rome, Italy, bohm@diag.uniroma1.it; Adriano Fazzone, CENTAI, Italy, adriano.fazzone@gmail.com; Stefano Leonardi, Sapienza, University of Rome, Italy, leonardi@diag.uniroma1.it; Cristina Menghini, Brown University, USA, cristina_menghini@brown.edu; Chris Schwiegelshohn, Aarhus University, Denmark, cschwiegelshohn@gmail.com.

2476-1249/2018/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

CCS Concepts: • **Theory of computation → Graph algorithms analysis**; • **Information systems →** *Web searching and information discovery*.

Additional Key Words and Phrases: densest subgraph; clustering; fairness; spectral graph analysis

ACM Reference Format:
Aris Anagnostopoulos, Luca Becchetti, Matteo Böhm, Adriano Fazzone, Stefano Leonardi, Cristina Menghini, and Chris Schwiegelshohn. 2018. Fair Projections as a Means Towards Balanced Recommendations. *Proc. ACM Meas. Anal. Comput. Syst.* 37, 4, Article 111 (August 2018), 31 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Recommender systems are prevalent in most content-providing online systems, as they increase the user satisfaction and, as a result, the revenue of the online service. All of the classical approaches, either based on content or on collaborative filtering, have the tendency to provide recommendations that are similar to the content that the user has previously followed. Clearly, this is desired and what makes these systems work. Of course, variants exist either to take care of new content or to allow exploration into the content space, with the hope that the user will discover some unknown to her topics.

Whereas this approach is in general desirable, there exist scenarios where it can also have some negative consequences. Consider a user browsing youtube and searching for content (talks, interviews, podcasts, etc.) on some controversial topic. For instance, "does the gender-pay gap exist?" or "should citizens be allowed to have guns?" There are arguments towards both sides of the spectrum. Yet, these are topics that are typically polarized, usually based on the individuals' prior beliefs. These beliefs typically get even more polarized as the user is exposed to content that supports her opinion. Suggestions from recommender systems, naturally lead to this, enforcing the creation of filter bubbles [37].

Diversity in recommender systems has been one of the design goals, as it leads to improved user experience [43, 44]. The survey by Kunaver and Požrl [43] describes various ways to define diversity (e.g., novelty, serendipity, etc.) and approaches in achieving it. Yet, the implicit goal of these approaches is to increase the user experience, eventually measured, for instance, by the number of completed transactions (clicks, purchases, etc.).

We want to propose a different objective. We propose *balanced recommendations*, which are recommendations that cover all the spectrum of a controversial topic. We assume that the content is classified according to a given dimension (e.g., conservative–liberal) and the goal is, when providing recommendations, to cover both sides. In the previously mentioned youtube example, the goal would be for the user to be exposed to content also of the opposite side. Such exposure has often the effect to increase critical thinking and moderate strong opinions [56]. This may be a tradeoff, as such an exposure may reduce the number of transactions. Other applications include diversification in friendship suggestion, in product recommendation as a means towards serendipity, by allowing to *explore* than to merely *exploit*. This holds true in product/information recommendations, but also in dating applications, social recommendations, and so on. In the long run, it can mitigate the "rich-get-richer" effect, by increasing the diversity in the recommendations provided.

Such a goal is complex: it requires the classification of content among various dimensions, the definition of a desired output, and the incorporation of balancing into existing recommendation approaches. In this work we start by defining the concept of balancing and showing how it can be achieved in some of the primitives that are being used in some recommender systems: clustering (captured by the k-clustering problem) and graph community detection (captured by the problem of maximum densest subgraph). Clustering in recommender systems has multiple applications: It is being used for user segmentation, for item categorization, for dimensionality reduction, and often to improve collaborative-filtering approaches by increasing the density of the user–item matrix. Furthermore, it can be used to improve scalability, and is a nice solution for the cold-start problem. The densest-subgraph primitive can also be a component in recommender systems based on graphs. As an example, we

mention community-based recommendations, as it can allow to identify users with similar interests. In the same lines, it can be used to provide recommendations in content-similarity graphs or in interest graphs. Similar to clustering, it can also help for the cold-start problem.

1.1 Contributions

In this paper we define mathematically the concept of balancing that we use and we provide some theoretical results towards what is achievable and what is not.

Densest Subgraph. As it turns out (see Section 3), the fair densest subgraph problem is intractable in general, whereas its unconstrained counterpart can be solved optimally through network flow [31]. Nevertheless, we have some quantifiable results regarding approximation algorithms in special cases. We can show that, if the underlying graph itself is fair, there exists a 2-approximation algorithm. We further show that, assuming the widely used small set expansion hypothesis [51], this is the best possible. We also consider the case where the graph itself is not fair and we instead aim for a proportional representation. For this, in our opinion more flexible variant of the problem, we show that the results for fair graphs can be extended.

Although this worst-case behavior is discouraging, the possibility of effective algorithms on practical instances is not ruled out. To this end, we identify properties that, if satisfied by some subgraph of the network under consideration, will afford recovery of an approximately fair, dense subgraph. More precisely, our goal in this respect is to design a heuristic that

(a) has a quantifiable guarantee if the underlying graph is well-behaved and
(b) is practically viable.

Our main result is a spectral algorithm that satisfies both of these requirements. In particular, the practical viability of our algorithm underscores that our notion of a well-behaved graph is a realistic one. As a candidate application, we consider the scenario of providing diverse recommendations of high quality, using data from the Amazon product co-purchasing graph. Our experiments not only confirm the quality of the output solutions, but also the scalability of our approach, which may not be the case for a conventional combinatorial approximation algorithm.

k-Clustering. We also study fair clustering problems. First, we show that computing a 3-approximation to the fair k-center clustering, which consists of minimizing the maximum distance in every cluster, is possible in *fixed-parameter tractable* running time, where the parameter in question is the number of clusters k. Similar results also hold for fair k-median and fair k-means. Notably, these results do not require an exponential dependency on the number of colors ℓ, which would happen with a naive application of coreset-based results. This is complemented with the following hardness proof. Given three point sets consisting of exactly n points, finding a fair n-median or n-center clustering is APX hard. This already shows that considering fair clustering with at least 3 populations is harder than the same problem with only 2 populations. In addition, approximating the fair k-center cost of a given candidate set of k points is similarly APX hard.

Overview of our approach. Our approach builds on the finding [36, 47] that the densest subgraph problem admits a spectral formulation. Specifically, an approximate densest subgraph can be computed by selecting nodes for inclusion according to the magnitudes of the corresponding entries in the main eigenvector of G's adjacency matrix. Unfortunately, this approach does not afford balanced solutions in general. In a nutshell, we sidestep this issue by first projecting the adjacency matrix onto a suitable "fair" subspace, an operation that corresponds to the enforcement of "soft" fairness constraints.

To see why the conventional spectral approach of [36] may not work[1] and why our approach mitigates the issue, Figure 1 presents plots obtained from Amazon books on US politics [41]. The books are labeled as either conservative or liberal, which corresponds to the labels −1 or 1. As described above, a candidate application may be to find a selection of books that are of interest to multiple readers, while mitigating potential polarization along political lines.

On the left, we observe the books ordered according to their corresponding entries in the main eigenvector of the adjacency matrix of the co-purchase graph. Books are also colored according to political orientation. We can observe that, whereas liberal books are well distributed, conservative ones are clustered. On the right we observe the results after application of our spectral embedding, which affords recovery of a subgraph of the co-purchase graph that is both dense and approximately balanced. Note that now conservative books are also well distributed along the principal component.

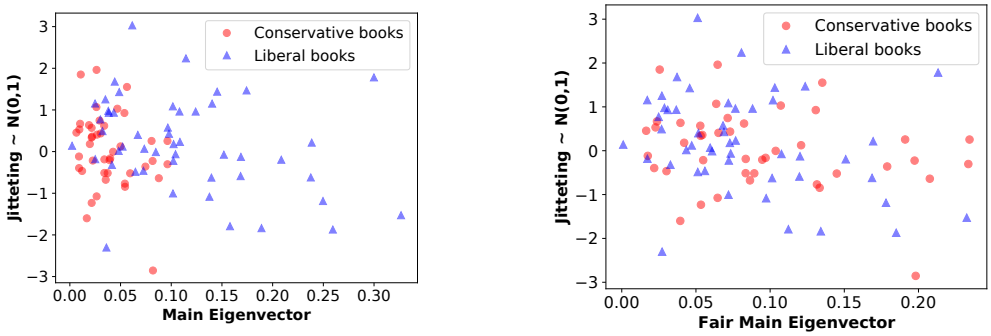


Fig. 1. Projection of books (see Section 5) onto the first principal component. (Left) Original data. (Right) Data after spectral embedding. Books are ordered on the x axis according to their corresponding entries in the main eigenvector, whereas on the y axis we have random noise for visualization.

1.2 Related Work

Apart from the vicinity of our work to the area of diversification in recommender systems that we discussed in the beginning of the introduction, there are various other areas that touch our work. In this section we describe some of them.

Densest Subgraph. Identifying dense subgraphs is a key primitive in a number of applications. Fratkin et al. [28], use it to identify over-represented but imperfectly conserved motifs in genomic DNA data, with the eventual goal of discovery of regulatory elements that determine the timing, location, and level of gene transcription. Gibson et al. [29] find dense subgraphs, as a means for web spam detection. Gionis et al. [30] make use of them in a means to improve efficiency in event sharing in social-networking sites. Some more applications, such as detection of genes that are correlated in some biological sense, are studied by Tsourakakis et al. [60], who present various variants of the densest subgraph problem. The standard problem can be solved optimally in polynomial time [31]. On the contrary, the fair densest subgraph problem is highly related to the densest subgraph problem limited to at most k nodes, which cannot be approximated up to a factor of $n^{1/(\log\log n)^c}$ for some $c > 0$ assuming the exponential time hypothesis [46] and for which state-of-the-art methods yield an $O(n^{1/4+\epsilon})$ approximation [13].

[1] In fact, this applies to any approach based on unconstrained maximization of the induced subgraph's density.

Algorithmic Fairness. Fairness in algorithms has received considerable attention in the recent past; see [34, 58, 61, 63] and references therein. The closely related notion of disparate impact was first proposed by Feldman et al. [27]. It has since been used by Zafar et al. [62] and Noriega-Campero et al. [50] for classification and Celis et al. [19, 20] for voting and ranking problems. Another problem that has received considerable attention is fair clustering. This was first proposed as a problem by Chierichetti et al. [22] in the case of a binary protected attribute. It was then investigated for various objectives and more color classes in theirs and subsequent work [5, 10, 11, 15, 18, 35, 52, 55].

Most closely related to our work are some recent works [40, 54, 57]. From those, the works of Samadi et al.[54] and Tantipongpipat et al. [57]. Consider the problem of executing a principal component analysis in a fair manner. Specifically, given a matrix A where the rows are colored (e.g., every row corresponds to a man or a woman), they ask for an algorithm that finds a rank-k matrix A' whose residual error $\|A - A'\|$ is small for both types of rows simultaneously. Whereas our method is similarly based on using the principal component in a fair manner, the difference is that we may be forced to treat the classes differently, if we aim to uncover a dense subgraph as illustrated in the example mentioned previously and illustrated in Figure 1.

The paper by Kleindessner et al. [40] considers spectral-clustering problems such as normalized cut. Like our work, they project the Laplacian matrix of a graph G onto a suitable "fair" subspace, and then run k-means on the subspace spanned by the smallest resulting eigenvectors. Under a fair version of the stochastic block model, they show that this algorithm recovers a planted fair partitions. Our work continues this idea by applying the technique to the densest subgraph problem.

Algorithmic Fairness and Fair Clustering. The idea of clustering using balancing constraints is based on the aforementioned notion of disparate impact [27]. Despite some impossibility results in certain settings [24, 38], it has been used successfully for classification [33, 50, 62], ranking [19, 20], regression [2], graph embeddings [17], team formation [8], and indeed clustering. Its application to clustering was initiated by Chierichetti et al. [22]. They showed that for two protected classes, fair clustering for various objectives such as k-median, k-center, and (implicitly, though unstated) k-means can be approximated as well as the unconstrained variants of the problem (up to constant factors). Building upon their work, [5, 18, 35, 55] considered this problem for large data sets. Backurs et al. [5] apply a technique of metric embedding to trees, to provide a close-to-linear approximation algorithm for the fair k-median problem. Huang et al. [35] construct carefully some *coresets*, which then combine with the techniques of Backurs et al. to accelerate solving k-median and k-means. A coreset is a data summarization technique, where a (weighted) point is a representative of part of the input. Coreset construction is also the method of choice of Braverman et al. [18], who use them to solve k-median under different distance metrics, and of Schmidt et al. [55] who compute them in a streaming computational model for k-means.

The main open problem left by Chierichetti et al. [22] is whether the approximability can be extended for multiple color classes. Here, the k-center problem has received the most attention with the current state of the art by Bercea et al. [12] being a 5-approximation algorithm, or a bicriteria 4-approximation algorithm that violates the fairness constraint by a small amount. For the special case of having equally sized color classes, Böhm et al. [15] gave a polynomial time constant factor approximation. Further results include a PTAS for constant k [35, 55] in Euclidean spaces of constant dimension and with 2 colors and bicriteria approximation algorithms [10, 12]. Other variants introduce capacity constraints and outliers [52] and overlapping clusters [10].

We note that there exist other models combining fairness and clustering objectives. Disparity of impact for spectral clustering has been studied by Kleindessner et al. [40]. Further spectral algorithms with fairness considerations appear in various other works [48, 54, 57]. Kleindessner et al. [39] considered k-centers under the fairness constraint that the set of centers, rather than the composition of the clusters.

1.3 Preliminaries and Notation

We consider an undirected graph $G(V, E, w)$, where V is the set of n nodes, $E \subseteq V \times V$ is the set of edges, and $w : E \to \mathbb{R}_{\geq 0}$ is a weight function. We denote the (weighted) adjacency matrix of G by A. For a subset $E' \subset E$ of the edges, we let $w(E') = \sum_{e \in E'} w(e)$. For unweighted graphs we have $w(e) = 1$ for each $e \in E$. For a node $u \in V$, its (weighted) degree (often called volume) is $d_u = \sum_{e \cap \{v\} \neq \emptyset} w(e)$. We also let $d_{\max} = \max_{u \in V} d_u$. For a $S \subseteq V$, we denote by G_S the induced subgraph. The *density* $D_S(G)$ of $S \subseteq V$ is the average degree of G_S, namely $D_S(G) = \frac{2 \cdot w(E \cap S \times S)}{|S|}$. We omit G from $D_S(G)$, whenever it is clear from context.

A *coloring* of the vertices is a map $c : V \to [\ell]$ of V, where $[\ell] := \{1, 2, \ldots, \ell\}$. A set $S \subseteq V$ is called *fair* or *balanced* if $|S \cap \{v \in V : c(v) = 1\}| = |S \cap \{v \in V : c(v) = 2\}| = \cdots = |S \cap \{v \in V : c(v) = \ell\}|$. A graph is called fair if V is fair. In the remainder, we provide positive results for the important case $\ell = 2$. In this case, for simplicity of exposition, we denote the colors *r*ed and *b*lue and we use $Red := \{v \in V : c(v) = red\}$ and $Blue := \{v \in V : c(v) = blue\}$ to refer to nodes of the respective color. To emphasize that the colors are disjoint, we write $V = \bigcup \uplus_{i=1}^{\ell} V_i$, where $V_i = \{v \in V \mid c(v) = i\}$

Definition 1.1 (Fair Densest Subgraph Problem). Given a (weighted) graph $G(V, E, w)$ and a coloring c of its vertices, identify a fair subset $S \subseteq V$ that maximizes D_S.

The fair densest subgraph problem is obviously a constrained version of the densest subgraph problem. It turns out to be considerably harder than its (polynomially solvable) unconstrained counterpart, as we show in Section 3.

Linear algebra notation. We denote by $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ the eigenvalues of A and by v_i A's ith eigenvector. We also set $\lambda = \max\{\lambda_2, |\lambda_n|\}$. Note that we always have $\lambda_1 \leq d_{\max}$. For a subset $S \subseteq V$, we denote by χ its normalized indicator vector, where S is understood from context. Namely, $\chi_i = 1/\sqrt{|S|}$ if $i \in S$, $\chi_i = 0$ otherwise. Finally, for a vector $x \in \mathbb{R}^n$, we let $\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2}$, the 2-norm of x.

Definition 1.2 (Clustering Problem). Given a set of n points P in some metric space and set of potential center sites F, the (k, z) clustering problem consists of computing a set $C \subset F$ of k center points in the metric space and an assignment $c : P \to C$ such that $\sqrt[z]{\sum_{p \in P} dist^z(p, c(p))}$ is minimized. Special cases include $z = 1$, which is k-median, and $z \to \infty$, which is k-center.

We define a *coloring* of the points in P similarly to the previous definition of a coloring of the vertices of a graph. Analogously, we extend the *balancedness* or *fairness* definitions to subsets of points of P. Now we can define the fair version of the clustering problem.

Definition 1.3 (Fair Clustering Problem). The fair-(k, z) clustering problem is the clustering problem that further requires that the set of points assigned to every center is balanced.

Given two n-point sets $A^{(i)}$ and $A^{(j)}$, a matching is a bijection $\pi : A^{(i)} \to A^{(j)}$. Given some matching π, we say that the z-cost is $\sqrt[z]{\sum_{x \in A^{(i)}} dist^z(x, \pi(x))}$. The optimal matching with respect to the z-cost is called the min-cost perfect z-matching, or simply min-cost perfect matching if z is clear from the context. In literature, this is sometimes referred to as the Earth Mover's distance between $A^{(i)}$ and $A^{(j)}$, for which we use the shorthand $EMD(A^{(i)}, A^{(j)})$. The time required to compute an optimal min-cost perfect matching on n-point sets is denoted by $MCPM(n)$[2].

[2]There exist algorithms that run faster in special cases, such as (2)-matching in low-dimensional Euclidean space. For a single algorithm that solves the problem for all z, we refer the reader as an example to the Hungarian algorithm [42].

2 SPECTRAL RELAXATIONS FOR THE FAIR DENSEST SUBGRAPH

As observed by Kannan and Vinay [36], the densest subgraph problem admits a spectral formulation. In particular, if we let x be an indicator vector over the vertex set, then the indicator vector of the vertex subset maximizing density is the maximizer of the following expression $\max_{x\in\{0,1\}^n} \frac{2x^TAx}{x^Tx}$.

Now, assume that each node is colored with one of two colors, red or blue. In the optimal solution x^* one of the colors might be overrepresented. To formulate the problem of computing a fair solution, we can add the constraint

$$\sum_{\text{node } i \text{ is red}} x_i = \sum_{\text{node } i \text{ is blue}} x_i \Leftrightarrow \quad \sum_{\text{node } i \text{ is red}} x_i \; - \sum_{\text{node } i \text{ is blue}} x_i = 0.$$

If we define the (unit 2-norm) vector $f_i = \begin{cases} \frac{1}{\sqrt{n}} & \text{if node } i \text{ is red} \\ -\frac{1}{\sqrt{n}} & \text{if node } i \text{ is blue,} \end{cases}$ the above constraint can be described as $f^Tx = 0$. We call such an x *fair*. Conversely, very unbiased solutions will have high, in absolute value, inner products with f.

Fair Densest Subgraph: Spectral Relaxation. Based on the considerations above, our approach transforms the input data (in this case the adjacency matrix A) by first projecting them onto the kernel of f. Namely, we first consider the following formulation of the fair densest subgraph problem: $\max_{x\in\{0,1\}^n} \frac{2x^T(I-ff^T)A(I-ff^T)x}{x^Tx}$. Note that, for any fair subset S with indicator x, we have $\frac{2x^TAx}{x^Tx} = \frac{2x^T(I-ff^T)A(I-ff^T)x}{x^Tx}$. Conversely, for any indicator vector $x \notin span(I - ff^T)$, the objective value can only decrease after we project to the kernel of f.

We next note that, by the discussion in the beginning of the section, by relaxing x to be an arbitrary vector, the above expression is maximized by the main eigenvector of $(I-ff^T)A(I-ff^T)$. The above relaxation corresponds to replacing hard fairness constraints with soft ones.

It is straightforward to encode more complicated fairness constraints using this technique. Suppose, for example, that we are given ℓ colors, and wish to output a subgraph such that every color is featured equally often. This induces a set of constraints $\sum_{\text{node } i \text{ is red}} x_i = \sum_{\text{node } i \text{ is blue}} x_i$, $\sum_{\text{node } i \text{ is red}} x_i = \sum_{\text{node } i \text{ is green}} x_i$, ... for all colors. The vectors satisfying all of these constraints lie in the nullspace of some $\ell - 1$ dimensional subspace S. Assume that F is a matrix such that the columns of F form an orthogonal basis of S. Then the above technique leads to the problem $\max_{x\in\{0,1\}^n} \frac{2x^T(I-FF^T)A(I-FF^T)x}{x^Tx}$. More generally, this technique can be extended to any system of linear constraints. One only has to merely find a suitable basis and project A onto said basis.

We note that even though the technique can handle these more complicated constraints, leveraging this in an algorithm with provable guarantees seems very difficult. Nevertheless, our experiments dealing with multiple colors showcase that we can still tackle more complicated fairness constraints with success in practice; see Section 5.1.

2.1 Recovery of Dense Fair Subgraphs in Almost Regular Graphs

In this section we provide some theoretical results, under some conditions on the input. Even though these conditions may not always be satisfied in practice, nevertheless the results provide evidence of the soundness of our modeling and solution approach.

We first need the following definition:

Definition 2.1. Graph $H = (V_H, E_H)$ is (d, ϵ)-regular if a d exists, such that $(1 - \epsilon)d \leq d_i \leq (1 + \epsilon)d$, for every $i \in V_H$.

Theorem 2.2. *Assume we have a graph $G = (V, E, w)$ with a 2-coloring of the nodes. Assume the spectrum of A satisfies $\lambda_1 \geq 4\lambda$.*[3] *Assume further that G contains a fair subset S such that: (1) G_S is (d, ϵ)-regular and (2) $d \geq (1-\theta)d_{\max}$, with $\epsilon + \theta \leq 1/4$. In this case, it is possible to recover all but $32(\epsilon+\theta)|S|$ of the vertices in S in polynomial time.*

Proof of Theorem 2.2. See Appendix A.1 for the proof. □

Intuitively, the result above states that, if the underlying network G is an expander containing an almost-regular, dense, and fair subgraph, we can approximately retrieve it in polynomial time. Succinctly, this follows because, under these assumptions, the indicator vector of S forms a small angle with the main eigenvector of $(I - ff^T)A(I - ff^T)$, which in turn has a large component on the main eigenvector of A. In particular, we run Algorithm GSA (see Algorithm 1) with $M = (I - ff^T)A(I - ff^T)$ and $\Delta = 16(\epsilon + \theta)$.

Algorithm: General Sweep Algorithm (GSA)
Data: Non-negative $n \times n$ matrix M, parameter Δ
Result: Subset $S \subseteq V$
$\hat{S} = \emptyset$; $\hat{D} = 0$;
Compute v_1 = main eigenvector of M;
Sort nodes $i \in V$ in non increasing order of $v_1(i)$;
`// Assume w.l.o.g. that {1,...,n} is resulting ordering of nodes in V;`
for $s = 1$ *to* n **do**
  $S = \{1, \ldots, s\}$
  Compute D_S = density of the subgraph induced by S
  if $D_S > \hat{D}$ *AND* $||S \cap Red| - |S \cap Blue|| \leq \Delta|S|$ **then**
    $\hat{S} = S$; $\hat{D} = D_S$
  end
end
return $\hat{S}$

Algorithm 1: General Sweep Algorithm (non-increasing).

The running time of Algorithm 1 is dominated by computing the first eigenvector and the projecting of the rows of the Laplacian onto said eigenvector. This can be done, up to $(1+\varepsilon)$ precision, in linear time.

3 HARD CONSTRAINTS AND HARDNESS OF APPROXIMATION

In general, enforcing fairness can make an easy problem intractable and this is the case for the densest subgraph problem. In this context, spectral relaxations can be regarded as a way to mitigate this issue, by enforcing soft fairness constraints to virtually any problem that is amenable to an algebraic formulation.

Nevertheless, in some cases it might be important to assess the *price of fairness*, by comparing the achievable quality of fair solutions to that of solutions for the original, unconstrained problem. In this section, we complement our algorithmic treatment of fairness with hardness results and approximation algorithms for specific cases. Some proofs are omitted for the sake of space, but they are available in the full version of the paper.[4] Some of our hardness results are based on the *small set expansion hypothesis*, which we now describe.

[3] That is, G is an expander.
[4] https://arxiv.org/abs/1905.13651

Consider a d-regular weighted graph G and, for every $S \subset V$, denote by $\Phi(S)$ the *expansion* (or conductance) of S [51]: $\Phi(S) = \frac{w(E \cap (S \times V \setminus S))}{\min\{\sum_{v \in S} d_v, \sum_{v \in V \setminus S} d_v\}}$. Given two constants $\delta, \eta \in (0,1)$, the small set expansion problem [51] $SSE(\delta, \eta)$ asks to distinguish between the following two cases:

Completeness There exists a set of nodes $S \subset V$ of size $\delta \cdot |V|$ such that $\Phi(S) \leq \eta$.
Soundness For every set of nodes $S \subset V$ of size $\delta \cdot |V|$, $\Phi(S) \geq 1 - \eta$.

Our hardness proofs are based on the small set expansion hypothesis defined as follows.

Conjecture 3.1 (SSEH). *For every $\eta > 0$ there exists a $\delta := \delta(\eta) > 0$ such that $SSE(\eta, \delta)$ is NP-hard.*

Recall from Section 1.2 that, whereas the densest subgraph problem is polynomially solvable, the best approximation for the densest at-most-k subgraph problem is in $O(n^{1/4})$ [14] and cannot be approximated up to a factor of $n^{1/(\log\log n)^c}$ for some $c > 0$ assuming the exponential time hypothesis [46] . The next theorem implies that these inapproximability results for the densest at-most-k subgraph problem hold also for the fair densest subgraph problem, showing that fairness constraints can drastically affect hardness of this problem.

Theorem 3.2. *The densest fair subgraph problem is at least as hard as the densest at most k subgraph problem. Moreover, any α-approximation to the densest at-most-k subgraph is a 2α approximation to densest fair subgraph.*

Proof. Consider an arbitrary graph $G(V, E)$. We consider V to be colored red. Add k blue nodes with no edges. Then the density of the fair densest subgraph is, up to a multiplicative factor of exactly $\frac{1}{2}$, equal to the density of the densest-at-most-$2k$ subgraph. Conversely, running an algorithm for densest k-subgraph with $k = \min(|Blue|, |Red|)$, and balancing out the resulting subgraph in post processing decreases the density by at most a factor 2. (This latter part is explained in more detail in the following theorem). □

When the input graph G is itself fair, we can provide stronger bounds.

1 **Input:** Graph $G(V, E, w)$
2 Compute the densest subgraph S
3 W.l.o.g $|S \cap Blue| \geq |S \cap Red|$
4 While $|S \cap Blue| > |S \cap Red|$, add an arbitrary node $v \in Red \setminus S$ to S
5 Return S

Algorithm 2: Approximate Fair Densest Subgraph

Theorem 3.3. *Given a fair graph $G(V, E, w)$, Algorithm 2 computes a fair set $S \subset V$, such that $2D_S \geq OPT$, where OPT is the density of the fair densest subgraph.*

Proof. We refer to the set S computed after line 2, and 4 as S_1 an S_2, respectively. Because S_1 is the unconstrained densest subgraph, $D_{S_1} > OPT$. For S_2, we observe that $|S_2| = |S_1| + |S_1 \cap \text{Blue}| - |S_1 \cap \text{Red}| \leq 2 \cdot |S_1|$, hence $D_{S_2} = \frac{w(E_{S_2})}{|S_2|} \geq \frac{w(E_{S_1})}{2|S_1|} \geq \frac{OPT}{2}$. □

The running times of both algorithms depend on the running time of the subroutines used to compute dense subgraphs. Unconstrained dense subgraphs can be found by solving a linear program or by computing a max flow [21, 31]. A faster $(1+\varepsilon)$ approximation that runs in time $O(n\text{polylog}(n))$ also exits [7, 25].

For the densest k-subgraph problem, the currently best algorithm that computes an $O(n^{1/4+\varepsilon})$ approximation runs in time $n^{O(1/\varepsilon)}$ [13].

We conclude this section by showing that approximating the fair densest subgraph problem beyond a factor of 2 is at least as hard as solving $SSE(\eta, \delta)$. Therefore, barring a major algorithmic breakthrough, Algorithm 2 is

optimal. The proof is based on the following idea: In regular graphs, for a given set of nodes S, the expansion $\Phi(S)$ is related to the density of S. We can use this, so that, given a graph G, we can carefully construct a colored graph G' such that finding the optimal fair densest subgraph in G' gives an estimate of the largest-expansion node set in G.

Theorem 3.4. *If SSEH holds, computing a $(2-\varepsilon)$ approximation of the fair densest subgraph problem in fair graphs is NP-hard for any $\varepsilon > 0$.*

Proof. See Appendix A.2 for the proof. □

4 FASTER ALGORITHMS FOR FAIR k-MEDIAN AND k-CENTER

Fair k-Median. For fair k-median, we obtain an $(\alpha+2)$-approximation, albeit in a substantially faster running time. As mentioned by Backurs et al. [5], computing a min-cost perfect matchings is expensive and tends to dominate the running time of fair clustering. In their paper, they proposed an algorithm that computes an $O(d \log n)$-approximate fairlet decomposition for fair k-median in nearly linear time[5]. We illustrate how to obtain a linear time randomized algorithm (i.e running in time $\tilde{O}(\ell \cdot MCPM(n))$), assuming that every color has an equal number of points.

Let us briefly recall some relevant definitions and results by Chierichetti et al. [22] A *fairlet decomposition*, is a k'-clustering with typically $k' > k$ centers, for which, for each cluster (*fairlet*), a single point is used as a representative. Clustering the representatives and merging the fairlets then results in a fair clustering, for any value of k. Note that the existence of a fair k-clustering always implies the existence of a fair n-clustering, for any number of colors. Chierichetti et al. show that computing an optimal fair n-median is possible if we are given only two colors. Even though the same problem is APX-hard for three colors (see Proposition 4.2), the following theorem establishes that a randomly sampled color is always a 2-approximate fair n-median in expectation. Repeating the sampling process allows us to find a good n-median clustering with high probability. The pseudocode is given in Algorithm 3.

Input: Balanced point set $A = \biguplus_{i=1}^{\ell} A^{(i)}$ with $|A^{(i)}| = n$
for $i \in \{1, \ldots, \log 1/\delta\}$
  Sample $A^{(t)} \subset A$ uniformly at random
  $cost(i) \leftarrow 0$
  for $j \in \{1, \ldots, \log \ell\}$
    Compute (approximate) $EMD(A^{(t)}, A^{(j)})$ and add it to cost
Output $A^{(t)}$ with minimal cost

Algorithm 3: Fast randomized fair n-median clustering

Theorem 4.1. *Let A be an $\ell \cdot n \times d$ matrix, let $c : [\ell \cdot n] \to [\ell]$ be a balanced coloring of A. Given an algorithm that computes a β-approximate fair n-median clustering with 2-colors in time $T(n,d)$, there exists a randomized algorithm that computes a (2β)-approximate fair n-median clustering with ℓ colors. The algorithm runs in time $O(\ell \cdot T(n,d) \log 1/\delta)$ and succeeds with probability $1-\delta$.*

Proof. We will start by recalling the following fact that establishes the metric properties of the Earth Mover's distance.

[5]The dependency on d may be further reduced to $O(\log k)$ using dimension reduction techniques by Makarychev et al. [45]

Fact 1 (Rubner et al. [53], Appendix A). *Let (X, d) be a metric space with points X and distance function d. Then the Earth Mover's distance on a (weighted) point sets of equal size (or total weight) using d as a ground distance is a metric.*

Given ℓ n-point sets $A^{(1)}, \ldots, A^{(\ell)}$ lying on some metric space, the fair n-median problem consists in finding an n-point set B such that $\sum_{i=1}^{\ell} EMD(A^{(i)}, B) = \sum_{i=1}^{\ell} \sum_{j=1}^{n} \min_{\pi \in \Pi: A^{(i)} \to B} d(A_j^{(i)}, \pi(A_j^{(i)}))$ is minimized.

We now sample a point set $A^{(t)}$ uniformly at random. Then

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{\ell} EMD(A^{(t)}, A^{(i)})\right] &\leq \mathbb{E}\left[\sum_{i=1}^{\ell}\Big(EMD(A^{(t)}, B) + EMD(A^{(i)}, B)\Big)\right] = \sum_{i=1}^{\ell} EMD(A^{(i)}, B) + \sum_{i=1}^{\ell} \mathbb{E}[EMD(A^{(t)}, B)] \\ &= \sum_{i=1}^{\ell} EMD(A^{(i)}, B) + \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} \frac{EMD(A^{(j)}, B)}{\ell} = 2\sum_{i=1}^{\ell} EMD(A^{(i)}, B),\end{aligned}$$

where the inequality follows from Fact 1. Hence, a random point set is always a good candidate solution for an approximate fair n-median clustering, with probability at least $1/2$. Repeating the sampling process $\log 1/\delta$ times and picking the best one yields a 2-approximation with probability $1 - (1 - 1/2)^{\log(1/\delta)} = 1 - \delta$.

We now run the β-approximate computation of fair n-median with respect to every sampled color $A^{(t)}$. Let $\pi_{i,t}$ be the matching computed by this algorithm, for every $i \in [\ell]$. We then have

$$\sum_{i=1}^{\ell}\sum_{p \in A^{(i)}}^{n} d(p, \pi_{i,t}(p)) \leq \sum_{i=1}^{\ell} \beta \cdot EMD(A^{(i)}, A^{(t)}) \leq \sum_{i=1}^{\ell} 2\beta \cdot EMD(A^{(i)}, B).$$

□

We further complement these results by showing that a fairlet decomposition is APX-hard for $\ell \geq 3$, both in the case of k-center and k-median. In particular, we also show that computing a better than 2-approximate n-center clustering decomposition is NP-hard for $\ell \geq 3$. Note that this stands in contrast to the computability of an optimal fairlet decomposition for $\ell = 2$ colors proposed by [22].

Proposition 4.2. *Let A be a set of $\ell \cdot n$ points in a finite metric, $\ell \geq 3$, let $c : [\ell \cdot n] \to [\ell]$ be a balanced coloring of A. Then approximating fair n-center within a factor of 2 and approximation fair n-median within a factor of $\frac{96}{95}$ is NP-hard.*

Proof. See Appendix A.3 for the proof. □

Fair k-Center. We show that for the special case of k-center in finite metrics, we can compute a set of k-centers that induce a 3-approximate fair k-clustering. Moreover, this algorithm runs in nearly linear time. The algorithm is essentially the farthest first traversal that is well known to produce an optimal 2-approximation for unconstrained metric k-center [32].

Theorem 4.3. *Let A be a set of $\ell \cdot n$ points in a finite metric, let $c : [\ell \cdot n] \to [\ell]$ be a balanced coloring of A, and let k be an integer. There exists an $O(ndk)$ time algorithm that computes a set of k points $C \subset A$ such that there exists a 3-approximate fair k clustering using C as centers.*

Proof. We argue why the final set of k points C computed by the farthest first heuristic fulfills the desired criteria.

Let OPT be the cost of the optimal solution to the fair k-center problem. First, consider the case that every point of C is in a different optimal cluster. In this case, we may upper bound the cost of clustering to C by $2OPT$ via the triangle inequality. If C does not intersect with all clusters of the optimal clustering, there must be some

cluster that contains at least two points of C. Let i be the first iteration in which this occurs and denote by C_{i-1} the points collected so far and by c_i the added point. It then holds that $d(c_i, C_{i-1}) \leq 2OPT$.

By definition of c_i, we know that for any cluster O_j with center o_j that has an empty intersection with C_{i-1}, we have $d(o_j, C_{i-1}) \leq d(c_i, C_{i-1})$. Because the distance of any point $p \in O_j$ to o_j is at most OPT, we therefore have

$$d(p, C_{i-1}) \leq d(p, o_j) + d(o_j, C_{i-1}) \leq OPT + d(c_i, C_{i-1}) \leq OPT + 2OPT = 3OPT. \tag{1}$$

Finally, we argue why there exists a valid fair clustering with this bound. The union of two disjoint balanced clusters is a balanced cluster. Let $c_j \in O_j \cap C_{i-1}$, for any cluster O_j intersecting with C_{i-1}. Due to the triangle inequality, the cost of any point in O_j is now upper bounded by at most $2OPT$. We assign all the points of O_j to c_j. For any cluster O_j not intersecting with C_{i-1}, we assign the points of O_j to the center minimizing $\min_{c \in C_{i-1}} d(o_j, c)$. Due to Equation 1, the cost of any point in O_j is therefore upper bounded by $3OPT$. □

However, we remark that although we can guarantee the existence of a good clustering using C as centers, it seems hard to recover it while ensuring fairness. This stands in contrast to unconstrained clustering, where one can simply assign every point to its closest center. For the special case $\ell = 2$, a fair clustering may be recovered using flow-based techniques. For $\ell \geq 3$, deciding whether there exists a clustering with some cost, given a candidate set of centers, it is a hard problem.

Proposition 4.4. *Let A be a set of $\ell \cdot n$ points in some finite metric with a fair coloring $c : [\ell \cdot n] \to [\ell]$, let C be (a possibly optimal) set of k centers and let $t > 0$ be a parameter. Then deciding whether there exists a fair k-center clustering using C as centers with the range $[t, 3t]$ is NP-hard.*

Proof. See Appendix A.4 for the proof. □

Lastly, we briefly show how to derive a $(1+\varepsilon)$ approximation for fair k-clustering in Euclidean spaces if the number of centers is constant. This shows that a separation between the hardness of unconstrained clustering and fair clustering has to consider large values of k.

Theorem 4.5. *Let A be a set of points in Euclidean space and let k be a constant. Then there exists an algorithm that computes in time $O(n^{poly(k/\varepsilon)})$ a $(1+\varepsilon)$-approximation for fair k-median, fair k-means, and fair k-center.*

Proof. The high-level idea is similar to early polynomial time approximation schemes for unconstrained k-clustering [1, 6], with a few modifications to account for fairness. Assume we are given an oracle that (1) returns a set of k centers such that these centers form a $(1+\varepsilon)$-approximation and (2) returns the size of the clusters associated to these centers. If we have access to both, we can recover a clustering with the same approximation ratio by solving the following minimum-transportation problem. For every color, we construct an assignment as follows. Every input point p corresponds to a node v_p in a flow network. Every center c corresponds to a node u_c. These nodes are connected by a unit capacity edge. Furthermore, we have unit capacity edges from the source node to each v_p, as well as edges from the nodes u_c to the target node. These edges have capacity that are exactly the target size of the clustering. We now find a feasible flow such that the connection cost $\sum_v \sum_u f(v_p, u_c) \cdot d(p, c)$ is minimized, where $d(p, c)$ corresponds to the Euclidean distance between points p and c [6]. Finding a feasible flow can be done in polynomial time, moreover such a flow is integral, i.e. guaranteed to be a fair assignment.

To remove the oracle, we do the following. For (2), we observe that there are $O(n^k)$ different ways of selecting the sizes of the k clusters, given a ground set of n points. For (1), it is well known that for all of the considered objectives, there exist weak coresets of for a single center of size $\text{poly}(\varepsilon^{-1})$, see [6] and [1]. Weak coresets essentially satisfy the following property: Given a point set A, a weak coreset wrt to some objective is a subset of S of A such that a $(1+\varepsilon)$ approximation computed on S is a $(1+O(\varepsilon))$ computed on A.

[6] For k-means, we would have to use squared Euclidean distances. For k-center, we would use a threshold network that only connects nodes to centers that are within distance $(1+\varepsilon) \cdot OPT$ and find an arbitrary flow.

Hence, we can find a suitable set of $\text{poly}(k/\varepsilon^{-1})$ points from which to compute k candidate centers by enumerating all $\text{poly}(k/\varepsilon^{-1})$-tuples in time $n^{\text{poly}(k/\varepsilon^{-1})}$. □

5 EXPERIMENTAL ANALYSIS

Whereas worst-case bounds give guarantees for algorithm behavior, they do not provide the entire picture when studying the empirical behavior. Algorithm 2 is theoretically optimal (assuming that the underlying graph is fair) and therefore theoretically superior to the spectral recovery schemes. As we now describe, the empirical performance between these approaches paints the opposite picture.

Overview. To test the performance of our algorithms on real data we used publicly available datasets. For our experiments we used an Intel Xeon 2.4GHz with 24GB of RAM running Linux Ubuntu 18.04 LTS. All methods have been implemented in Python3 using the functionalities provided by NetworkX [7] and SciPy [8] libraries.

Datasets for Fair Densest Subgraph. The PolBooks dataset [41] is an undirected unweighted graph,[9] whose nodes represent books on US politics included in the Amazon catalog, and an edge between two books exists if both books are frequently co-purchased by the same buyers. Each book is further labeled depending on its political stance, possible labels being *liberal*, *neutral*, and *conservative*. For our experiments, we considered only the subgraph induced by *liberal* and *conservative* books, obtaining 92 nodes (43 of which were associated with a *conservative* world view, 49 with a *liberal* world view) for 374 edges in total.

The Amazon products metadata dataset [49] contains descriptions for 15.5 million Amazon products.[10] For a single product, we only considered the product id (*asin* field), the category the product belongs to (*main_cat* field), and the set of frequently co-purchased products (*also_buy* field). It should be noted that in this dataset, each node belongs to exactly one (main) Amazon category so that, together, these three fields allow recovery of a large, undirected, labeled graph, with products as nodes, categories as labels, and edges representing frequent co-purchasing product pairs. For this dataset, we leveraged the co-purchasing relation among products to naturally extract undirected and unweighted labeled graphs. In more detail, for each pair (ℓ_1, ℓ_2) of Amazon main categories, we extracted the undirected subgraph induced by the subset of nodes of category ℓ_1 (ℓ_2) that have at least one neighbor from category ℓ_2 (ℓ_1). We did not consider graphs with fewer than 100 nodes. In this way, we retrieved 299 subgraphs of two categories (colors), with sizes ranging between 103 and 33,922 nodes. We extended and applied this procedure to triples (ℓ_1, ℓ_2, ℓ_3) and quadruples $(\ell_1, \ell_2, \ell_3, \ell_4)$ of labels, obtaining 1,147 subgraphs of three categories (colors), with sizes ranging between 352 and 30,135 nodes, and 1,408 subgraphs of four categories (colors), with sizes ranging between 1,521 and 30,086 nodes.

The Facebook100 dataset [59] contains 100 anonymized undirected unweighted graphs constructed from the Facebook social network. Each graph is associated with a single university in the United States, where nodes represent Facebook accounts of people affiliated with the university, and edges represent friendship relations between these Facebook accounts. Nodes have several attributes, among which we selected *gender* (`male`, `female`, `not_specified`) and *profession* (dichotomized as `student`, `not_student`). We considered three versions of this dataset: for the two-color version, we considered the graphs induced by nodes with values for the *gender* attribute equal to `male` or `female`; for the three-color version, we considered also the value `not_specified` for this attribute; and for the four-colors version, we combined the *gender* and *profession* attributes, considering only graphs induced by nodes with the following values for these attributes: (`male`, `student`), (`male`, `not_student`),

[7] https://networkx.github.io/documentation/stable
[8] https://www.scipy.org
[9] http://www.casos.cs.cmu.edu/computational_tools/ datasets/external/polbooks/polbooks.gml.
[10] https://nijianmo.github.io/amazon/index.html

(female, student), (female, not_student). We generated 100 graphs of two and four colors, with sizes ranging between 701 and 38,786 nodes, and 100 graphs of three colors, with sizes ranging between 769 and 41,554 nodes.

Datasets for Fair k-Clustering. The six datasets used for experiments are taken from previous literature [5, 10, 22]. As our interest is in the multiple-color scenario, we ran our experiments considering 8 colors. Each color represents a protected class characterized by some particular value of the chosen protected attributes. For each dataset, we selected protected attributes to obtain 8 classes in total, and we also subsampled the original records to get the same number of records for each class. Each sample is a perfectly balanced set of points with respect to the eight colors described in the following.

The *Adults* dataset[11] contains "1994 US census" records about registered individuals including age, education, marital status, occupation, ethnicity, sex, hours worked per week, native country, and others. Following prior work [10] and [22], the numerical attributes chosen to represent points in the Euclidean space are AGE, FNLWGT, EDUCATION-NUM, CAPITAL-GAIN, and HOURS-PER-WEEK. The protected attributes chosen to represent the classes are SEX, ETHNICITY, and INCOME, where each of them takes only 2 possible values. For the experiments, we used 100 balanced subsamples of 1000 distinct records.

The *Athletes* dataset[12] contains information on Olympic athletes and medal results from Athens 1896 to Rio 2016. The selected features are AGE, HEIGHT, and WEIGHT. The protected attributes are SEX (Female, Male), SPORT (we selected two sports—gymnastics and basketball), and MEDAL (we considered two types of athletes for—athletes who won at least one medal and athletes who did not). For the experiments, we used 100 balanced subsamples of 1000 distinct records.

The *Bank* dataset[13] stems from direct marketing campaigns, based on phone calls, of a Portuguese banking institution. As in [10] and [22], the selected features to represent the points in the space are AGE, BALANCE, and DURATION. The protected attributes are MARITAL STATUS (Married or notMarried), EDUCATION (Secondary or Tertiary), and HOUSING (True or False). For the experiments, we used 100 balanced subsamples of 1000 distinct records.

The *Diabetes* dataset,[14] used for experiments in [22], represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes; of these features, 4 were chosen to represent the points in the space: TIME_IN_HOSPITAL, NUM_LAB_PROCEDURES, NUM_MEDICATIONS, and NUMBER_DIAGNOSES. The protected attributes are SEX (Female, Male), ETHNICITY (Caucasian or AfricanAmerican), and age (this attribute has been dichotomized in order to have two classes of ages: people who are respectively less and more than 50 years old). For the experiments, we used 100 balanced subsamples of 1000 distinct records.

The *Credit-Cards* dataset,[15] contains information on credit-card holders from a certain credit card in Taiwan. Here, we selected the same 14 features chosen by [10] and the protected attributes that we consider are SEX (Female, Male), EDUCATION (GraduateSchool or University), and MARRIAGE (married or notMaried). For the experiments, we used 100 balanced subsamples of 1000 distinct records.

The *CensusII* data set contains records extracted from the *USCensus1990raw*[16] data set (also used in [5]), containing 2,458,285 records composed of 68 attributes. Among all of these attributes, we chose 9 to represent the points in the Euclidean space: AGE, AVAIL, CITIZEN, CLASS, DEPART, HOUR89, HOURS, PWGT1, and TRAVTIME. For this dataset, the selected protected attributes are SEX (Female, Male), RACE (dichotomized as White, notWhite), and

[11] https://archive.ics.uci.edu/ml/datasets/Adult
[12] https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results
[13] https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
[14] https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008
[15] https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients
[16] https://archive.ics.uci.edu/ml/datasets/US+Census+Data+%281990%29

MARITAL (dichotomized as NowMarried, NowNotMarried). For the experiments, we used 100 balanced subsamples of 1000 and another 100 balanced subsamples of 450,000 distinct records. We refer to the latter collection of subsamples as the *CensusII 450K points* dataset.

Algorithms for Fair Densest Subgraph. We compare the performance of the following algorithms, which for simplicity we describe in the two-colors scenario:

2-DFSG. The optimal 2-approximation algorithm (Algorithm 2) based on Goldberg's optimal algorithm for the densest subgraph problem [31], described in Section 3.

Spectral Algorithms. Following prior work [36, 47] and Theorem 2.2, we ran a variety of eigenvector-rounding algorithms. These are all variants of a modified version of the General Sweep Algorithm (Algorithm 1) used in the proof of Theorem 2.2 that sorts the entries of the main eigenvector of M four times (instead of a single one) according to the following criteria: (1) nonincreasing; (2) nondecreasing; (3) nonincreasing absolute values; (4) nondecreasing absolute values. With these premises, we consider the following spectral algorithms: The first two are just the modified version of Algorithm 1 with different choices for M, whereas **PS** and **FPS** perform a slightly modified sweep that always affords a fair solution.

Single Sweep (SS). This algorithm is simply Algorithm 1, when all previously mentioned sorting criteria are used, with $M = A$ and $\Delta = 0$.

Fair Single Sweep (FSS). It is the execution of **SS**, this time on matrix $(I - ff^T)A(I - ff^T)$ instead of A.

Paired Sweep (PS). Paired Sweep is a modification of **SS** in which the fairness constraint is satisfied by construction in each subgraph produced by the rounding algorithm. This is done by considering the subsets *Red* and *Blue* of the nodes, sorting each of them separately according to the values of the corresponding entries in the main eigenvector of A and then, for each $s = 1, \ldots, \min\{|Red|, |Blue|\}$ considering the candidate set of nodes of cardinality $2s$ obtained by taking the first s nodes from each ordered subset. For a pseudocode, we refer to Algorithm 4.

Data: Graph $G(V, E)$, with $V = Red \cup Blue$, $n \times n$ adjacency matrix M, parameter Δ
Result: Subset $S \subseteq V$

$\hat{S} = \emptyset$; $\hat{D} = 0$;
Compute v_1 = main eigenvector of M;
Sort nodes $i \in Red$ and nodes $j \in Blue$ in non increasing order wrt v_1
// Assume w.l.o.g. that $\Pi_{\texttt{red}} = \{1, \ldots, |Red|\}$ and $\Pi_{\texttt{blue}} = \{1, \ldots, |Blue|\}$ is resulting ordering of nodes in V;
Fuse node i from Π_{red} with node i from Π_{blue}
for $s = 1$ *to* $\min(|Red|, |Blue|)$ **do**
    $S = \{1, \ldots, s\}$
    Compute D_S = density of the subgraph induced by S
    if $D_S > \hat{D}$ *AND* $||S \cap Red| - |S \cap Blue|| \leq \Delta|S|$ **then**
        $\hat{S} = S$; $\hat{D} = D_S$
    end
end
return $\hat{S}$

Algorithm 4: Paired Sweep Algorithm.

Fair Paired Sweep (FPS). It is the execution of **PS**, this time on matrix $(I - ff^T)A(I - ff^T)$ instead of A.

Algorithms for Fair k-Clustering. We solved the fair k-median problem by implementing Algorithm 3, algorithm **Q**, and algorithm **Excellent**. Algorithm **Q** is similar to the algorithm by Böhm et al. [15], except that we select the color with minimum perfect-matching cost. This variant is guaranteed to return an $(\alpha + 4)$ approximation and is slightly faster than the original algorithm in [15]. Algorithm **Excellent** is a further variant of the same original algorithm [15] that computes a good clustering for each color and subsequently performs a fair assignment. The approximation factor is theoretically equal to that of Böhm et al. [15] but empirically, as we show later, the results are often superior.

Given the strong relationship between **Q** and **Excellent** with the algorithm from [15], we also implemented the last one for comparison.

We ran the algorithms for all values of k between 2 and 20; for one center, any solution is naturally fair. We observed that that there was already little to no difference between cost of a fair 20-clustering and the cost of a fair n-clustering, so we did not consider larger values of k.

We compared these algorithms with the implementation of Bera et al. [10]. For the largest data set (*CensusII 450K points*) consisting of 8 colors with a total of 450,000 points, their code did not terminate. On this dataset, we showcased the modularity of our approach by combining it with the fast fairlet algorithm by Backurs et al. [5], something not possible with the approach of Bera et al. Because our algorithm requires a solver for the unconstrained k-median problem, for all 1000-points datasets, we used the single-swap local search strategy, while yields a 5-approximation in the worst case [4].

For the *CensusII 450K points* dataset, local search is infeasible to run. Instead, we used a simple heuristic that essentially mimics the k-means++ algorithm [3]: First we sample k centers by iteratively selecting the next center with probability proportional to its distance to the previously chosen centers, and then running the k-medoids algorithm to further refine the solution.

5.1 Results

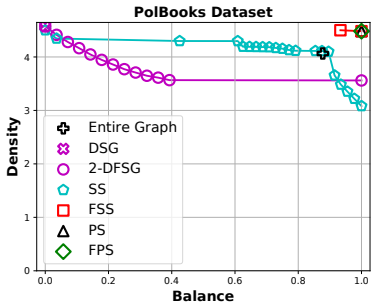
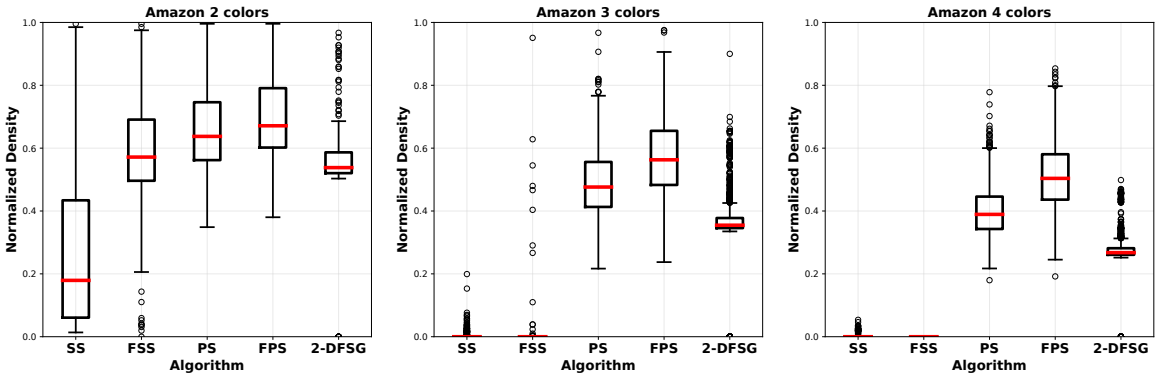


Fig. 2. Pareto front of the subgraphs generated by each algorithm, with respect to density and balance, on PolBooks dataset.

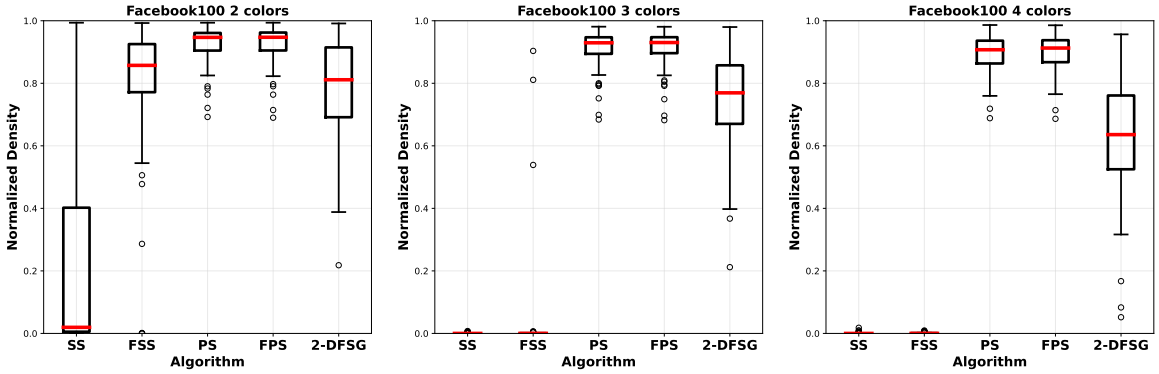
Fair Densest Subgraph. Figure 2 shows the performance of our algorithms on PolBooks dataset through the Pareto front of the subgraphs generated by each algorithm during its execution with respect to density and balance.[17] **PS** and **FPS** by construction only return fair solutions whereas the other algorithms potentially have tradeoffs. In particular, the **2-DFSG** (Algorithm 2) starts at the unconstrained optimum (**DSG**) and proceeds to add nodes that increase balance while potentially decreasing density.

[17] Given two color classes *Red* and *Blue*, we define the *balance* of a subgraph containing x *Red* and y *Blue* nodes as $\min\left(\frac{x}{y}, \frac{y}{x}\right)$.



(a) Amazon 2 colors: 299 subgraphs. (b) Amazon 3 colors: 1147 subgraphs. (c) Amazon 4 colors: 1408 subgraphs.

Fig. 3. Performance of our algorithms on Amazon dataset for 2, 3, and 4 colors on 299, 1147 and 1408 samples (subgraphs) respectively. Reported are aggregates over all generated subgraphs, with unfair solutions receiving a density of 0, see Table 1.



(a) Facebook100 2 colors: 100 subgraphs. (b) Facebook100 3 colors: 100 subgraphs. (c) Facebook100 4 colors: 100 subgraphs.

Fig. 4. Performance of our algorithms on Facebook100 dataset for 2, 3, and 4 colors on 100 subgraphs. Reported are aggregates over all generated subgraphs. The density values are normalized to $[0, 1]$, considering the density of the input graph as the minimum value and the density of the densest subgraph as the maximum value.

Figure 3 shows the distributions of the normalized density, over the entire set of Amazon instances (for two, three, and four colors), of the fair subgraphs retrieved by different algorithms. Normalization, performed to make solutions for different instances comparable, is done by scaling to the optimal density of the unconstrained problem, making the maximum possible value on the y-axis equal to 1. Experimental results represented in Figure 3(a, b, and c) show that spectral heuristics based on the paired-sweep technique (**PS** and **FPS**) consistently outperform **2-DFSG** algorithm, despite its theoretical optimality (proved in a two-color scenario and in presence of a fair input graph), regardless of the number of considered colors. In more detail, the **FPS** heuristic is the method that achieves the maximum median density. According to Figure 3(b and c), it is evident that for a number of colors greater than two, the spectral methods that do not rely on the paired-sweep technique (**SS** and **FS**) are not

the appropriate approaches for tackling the problem. Focusing on the two-color scenario, depicted in Figure 3(a), we have that, with the exception of **SS**, which uses the original adjacency matrix and whose distribution is skewed towards lower density values, the performance of spectral heuristics is comparable with that of **FPS** achieving the highest median density. Always in the two-color scenario, we can observe that algorithms run on $(I - ff^T)A(I - ff^T)$ (**FSS** and **FPS**) respectively outperform their counterparts (**SS** and **PS**) run on A.

We report in Table 1 the percentage of instances each algorithm is not able to solve, that is, for which it does not return a fair solution and, consequently, we assign a density equal to 0.

#Colors	#Samples	SS	FSS	PS	FPS	2-DFSG
2	299	0	0.33	0	0	3.01
3	1,147	73.93	95.55	0	0	5.31
4	1,408	92.54	99.64	0	0	1.91

Table 1. Percentages of unfair solutions for Amazon dataset.

The data reported in Table 1 confirm the observation that spectral methods that do not rely on the paired-sweep technique essentially fail to recover a dense fair subgraph in a context that involves more than two colors: the **SS** and **FSS** methods provided unfair solutions for almost all samples when the number of considered colors is greater than 2. As noted previously, **PS** and **FPS** cannot return unfair solutions: this is the reason behind the presence of zeros in their columns. It is worth to say that **2-DFSG** (Algorithm 2) results in an unfair solution if the original graph is unbalanced and the unconstrained densest subgraph cannot be made fair via line 4. This justifies the presence of quantities greater than zero in the last column.

Amazon	2 colors	3 colors	4 colors
#Samples	299	1,147	1,408
2-DFSG	46,388 (*101,391*)	151,049 (*152,898*)	127,834 (*75,276*)
FPS	360 (*659*)	1,083 (*2,073*)	745 (*524*)
PS	424 (*842*)	1,130 (*2,106*)	775 (*572*)
FSS	465 (*861*)	1,652 (*2,185*)	1,369 (*984*)
SS	463 (*859*)	1,665 (*2,216*)	1,368 (*986*)

Table 2. Average and standard deviation of the running times (in msec) of all proposed methods on Amazon dataset: 2, 3, and 4 colors.

Facebook100	2 colors	3 colors	4 colors
#Samples	100	100	100
2-DFSG	373,092 (*361,407*)	400,628 (*364,886*)	363,625 (*353,587*)
FPS	1,354 (*1,338*)	1,815 (*1,342*)	1,892 (*1,354*)
PS	1,348 (*1,346*)	1,219 (*1,042*)	1,141 (*1,002*)
FSS	1,159 (*1,001*)	1,773 (*1,278*)	1,829 (*1,259*)
SS	1,131 (*980*)	1,179 (*988*)	1,089 (*921*)

Table 3. Average and standard deviation of the running times (in msec) of all proposed methods on Facebook100 dataset: 2, 3, and 4 colors.

To show the versatility of our methods, we replicated the experiment performed with the Amazon dataset on the Facebook100 dataset while employing a slightly different notion of fairness. Here, we consider a set of nodes to be fair if, for all pairs of colors, the ratio of the number of nodes belonging to these colors in the set equals the corresponding ratio in the original graph. This type of constraint represents a linear dependence between colors, and it is directly incorporated into our spectral embedding as discussed in Section 2. Additionally, Algorithm 2 can be easily modified to handle this type of constraint by adapting line 4 accordingly. As any input graph satisfies this definition of fairness, all our methods provide fair solutions in output. Because of this fact, in this experiment the density values of solutions provided by our methods are normalized in $[0, 1]$, considering the density of the input graph as the minimum value and the density of the densest subgraph as the maximum value. Figure 4 reports results for this experiment on the Facebook100 dataset. Results are similar to those for Amazon dataset.

Tables 2 and 3 report that spectral methods are faster than **2-DFSG**. Indeed, the average running time of the **2-DFSG** method is of two orders of magnitude greater than the one required by the spectral methods. This is

Amazon dataset	2 colors	3 colors	4 colors
#Nodes, #Edges	108,230/1,851,733	108,185/1,132,578	108,220/1,360,241
2-DFSG	4,126,002 /*0.50*	3,618,960 / *0.34*	3,991,358 / *0.27*
FPS	36,199 / *0.65*	11,467 / *0.45*	31,988 / *0.61*
PS	91,582 / *0.56*	39,327 / *0.45*	32,643 / *0.50*
FSS	33,074 / *0.51*	17,358 / *NoFairSol*	45,465 / *NoFairSol*
SS	26,429 / *0.21*	24,161 / *NoFairSol*	32,324 / *NoFairSol*

Table 4. Running time (in msec) and solution quality (expressed as normalized density of the retrieved fair subgraph to the optimal density of the unconstrained problem) of all proposed methods on three Amazon subgraphs with 2, 3, and 4 colors each. Each subgraph has roughly 100K nodes and 1.1M edges.

coherent with the fact that the **2-DFSG** method requires solving the Max-Flow problem, which is computationally expensive.

Table 4 reports execution time and solution quality of all proposed methods on three not small-sized Amazon subgraphs with 2, 3, and 4 colors each. In particular, for what concerns the quality of the provided solutions, the results provided in Table 4 are completely in line with with the information extracted from Figure 3 and Table 1. The relation among execution times are also in line with what provided in Table 2. Moreover, we can see that on the considered instances (2, 3, and 4 colors, 100K nodes and 1.1M edges) the **2-DFSG** method requires slightly more than one hour of computation, against 91sec required by the paired spectral heuristics (**PS** and **FPS**). These results suggest that the spectral approaches are suitable for dealing with not small-sized graphs.

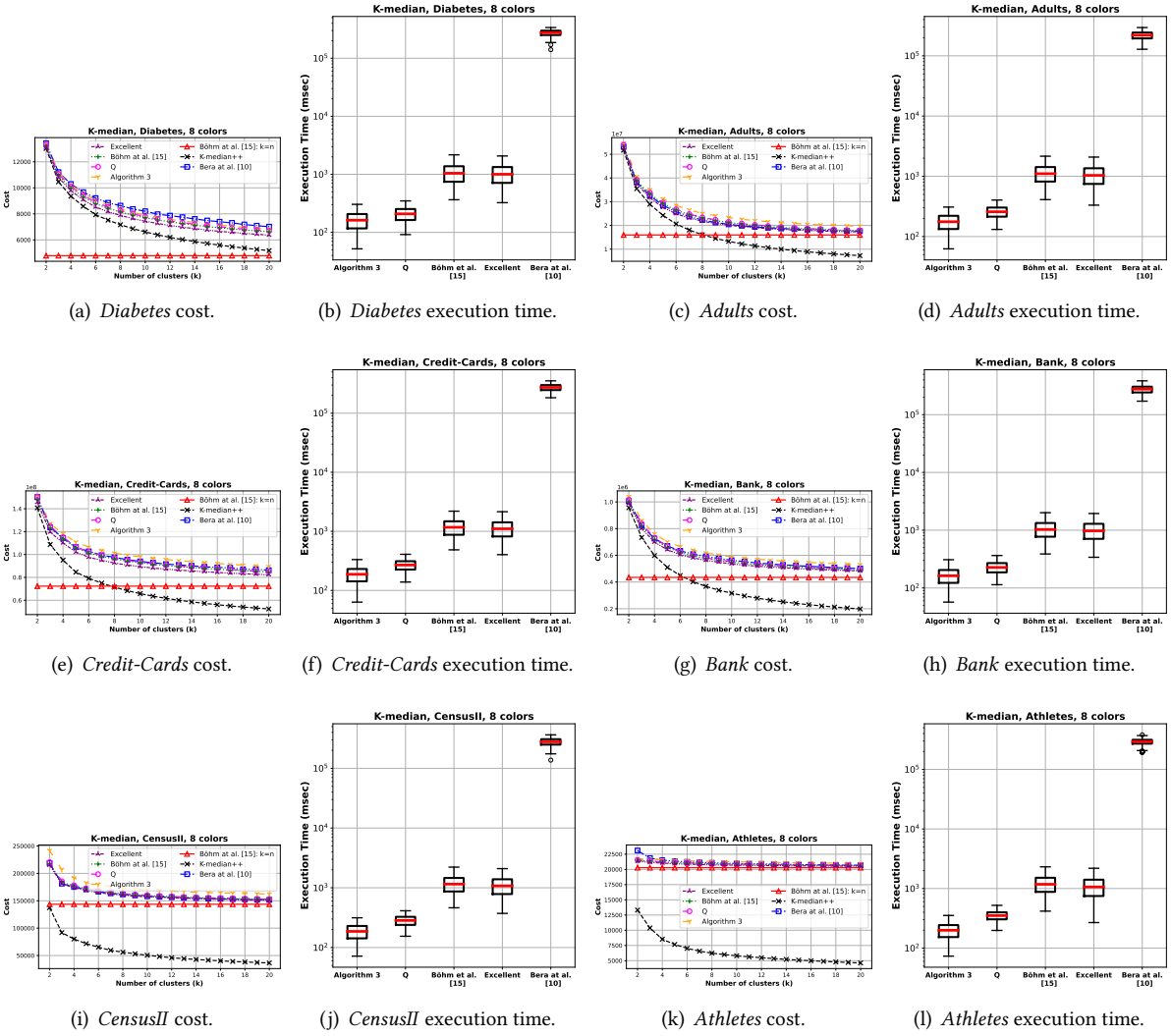
The experiments also provide information on the performance of both spectral and combinatorial approaches designed for the unconstrained problem on the densest fair subgraph problem. In particular, the performance of the spectral algorithm for the unconstrained problem based on [36, 47] cannot be better than those of **SS** on the constrained version of the problem. This is because both methods generate the same set of candidate solutions, and, while **SS** provides in output the densest fair solution among the candidate solutions (if any), the method for the unconstrained problem always provides in the output the densest one, that is not necessarily a feasible one. Furthermore, the performance of **2-DFSG** provides information on the fact that the global optimum to the unconstrained problem provided by the Goldberg's algorithm [31] never provided a feasible solution for the densest fair subgraph problem. This is shown in Figure 3, Figure 4, and Table 4, where **2-DFSG** never provide solutions with maximum normalized density.

These results highlight the significant shortcomings of algorithms from the unconstrained problem in addressing the densest fair subgraph problem.

Fair k-Clustering. In this experimental analysis, we compare the performance of algorithms for the fair k-median problem in terms of both solution cost and execution time. We additionally report the cost of an unconstrained solution and the cost for a fair n clustering solution, which provides a lower bound for the cost of any fairlet-based algorithm.

Regarding solution cost (Figure 5(a), 5(c), 5(e), 5(g), 5(i), and 5(k); Table 5, 6, 7, 8, 9, and 10), across all tested datasets of 1000-points, all algorithms perform slightly better than the one of Bera et al. [10] on instances in which the fairlets (i.e., the fair n clusterings) are very cheap compared to the cost of a k-clustering. This cost difference shrinks on datasets where fairlets are more expensive. **Excellent** consistently produces the cheapest feasible solutions among all algorithms, on average, and the average solution cost of **Q** is comparable to that of the algorithm by Böhm et al. [15], which never exceeds the cost of Algorithm 3.

In terms of running time (Figure 5(b), 5(d), 5(f), 5(h), 5(k), and 5(l)), all algorithms run substantially faster than the one by Bera et al. [10] by roughly factors of 100 or more. Algorithms 3 and **Q** report an average running time of 176msec and 258msec, respectively. This is also significantly faster than the algorithm by Böhm et al.![15] and **Excellent** (1111msec and 1044msec on average respectively), while having a roughly comparable cost.



(a) *Diabetes* cost. (b) *Diabetes* execution time. (c) *Adults* cost. (d) *Adults* execution time.

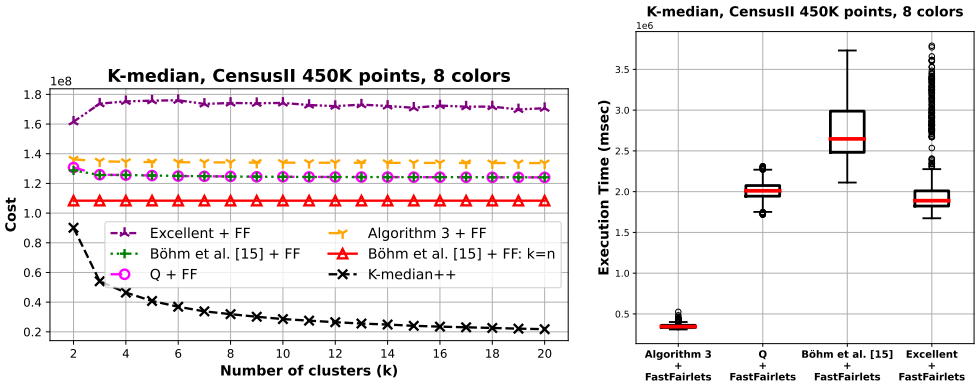
(e) *Credit-Cards* cost. (f) *Credit-Cards* execution time. (g) *Bank* cost. (h) *Bank* execution time.

(i) *CensusII* cost. (j) *CensusII* execution time. (k) *Athletes* cost. (l) *Athletes* execution time.

Fig. 5. Average cost and execution time of the fair-k-median algorithms on all datasets of 1000 points: 8 colors and 100 subsamples for each dataset.

Furthermore, we empirically observe that the algorithm of Bera et al. [10] almost always computes a balanced solution, as opposed to the bicriteria result stated in their paper. Specifically, fewer than 0.8% of the instances for *Diabetes*, fewer than 0.6% of the instances for *Credit-Cards* dataset, fewer than 0.3% of the instances for *Adults* dataset, fewer than the 0.2% of instances for *Athletes* dataset, and fewer than the 0.05% of instances for *Bank* yielding an unfair solution. By design, our algorithms and the algorithm by Böhm et al. [15] always guarantee fairness; on the other hand, solutions returned by the unconstrained k-median method are highly unbalanced.

For the larger 450,000-point variant of the *CensusII* data set (*CensusII 450K points*), we use the fast fairlet decomposition by Backurs et al. [5] to ensure scalability. Unfortunately, the implementation by Bera et al. [10]



(a) *CensusII 450K points* cost.

(b) *CensusII 450K points* execution time.

Fig. 6. Average cost and execution time of the fair-k-median methods combined with fast fairlets decomposition [5] on the *CensusII 450K points* dataset: 8 colors, 100 subsamples of 450,000 distinct points each. The algorithm of Bera et al. [10] is missing as it was not able to terminate in this dataset.

could not benefit from this preprocessing step, and the implementation itself is not able to process data sets at this scale. Relations among running times of algorithms are coherent to those from the smaller data sets (Figure 6(b)). In general, the most notable difference is that computing an approximate fair assignment after optimization as done by **Excellent** negatively affects the approximation (Figure 6(a) and Table 11).

6 FUTURE WORK

Future work might consider extending the spectral approach to more involved fairness constraints with provable guarantees. Empirically, we already observed that, although the spectral algorithms retain a good behavior both theoretically and empirically, the performance of the approximation algorithm deteriorates. We identify two key problems that may be more manageable. First, one might consider the case where the graph only has two colors, but the colors may overlap, that is, a node can be both red and blue. Clearly, the approximation results still hold in this case. Can one improve the analysis of the spectral recovery scheme, depending on the degree of overlap? Second, one might consider the case of multiple disjoint colors, each of equal size. Such considerations have been studied in clustering literature [9, 10, 16, 26]. Is it possible to derive similar results for densest subgraph?

For clustering, a number of problems are left in this work. The most challenging one is to show whether the constant factor loss in the approximation is necessary or not. In other words, does there exist a result showing that fair clustering is strictly harder than unconstrained clustering, for any objective? Since this question is rather general and might be hard to answer, we propose a few simpler problems. First, we have shown that in general metrics, fair n-center is APX-hard if the number of colors is greater than 3. Does this result also hold for the Euclidean plane? Moreover, what can we say about computing a fair n-median? We also showed that a PTAS for fair clustering exists, provided that k is constant. k-median and k-means in constant dimension admit a PTAS. A natural question is whether such a PTAS also exists for the fair variants of the problem. This problem is open, even in the case of two protected attributes.

Of course, solving the two problems that we study in this paper does not give a recommender system that exposes the user to a wide range of topics. The main challenge is to use the ideas of this paper and apply them in graph-based or clustering-based approaches. Plainly using these algorithms instead of classical approaches

inside recommender systems will almost certainly not work, so a lot of experimentation is required to arrive at a working system.

REFERENCES

[1] Marcel R. Ackermann, Johannes Blömer, and Christian Sohler. 2008. Clustering for metric and non-metric distance measures. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*. 799–808.

[2] Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. 2019. Fair Regression: Quantitative Definitions and Reduction-Based Algorithms. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. 120–129.

[3] David Arthur and Sergei Vassilvitskii. 2007. k-means++: the advantages of careful seeding. In *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 1027 – 1035.

[4] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. 2004. Local Search Heuristics for k-Median and Facility Location Problems. *SIAM J. Comput.* 33, 3 (2004), 544–562.

[5] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. 2019. Scalable Fair Clustering. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. 405–413.

[6] Mihai Badoiu, Sariel Har-Peled, and Piotr Indyk. 2002. Approximate clustering via core-sets. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*. 250–257.

[7] Bahman Bahmani, Ashish Goel, and Kamesh Munagala. 2014. Efficient Primal-Dual Graph Algorithms for MapReduce. In *Algorithms and Models for the Web Graph - 11th International Workshop, WAW 2014, Beijing, China, December 17-18, 2014, Proceedings*. 59–78. https://doi.org/10.1007/978-3-319-13123-8_6

[8] Giorgio Barnabò, Adriano Fazzone, Stefano Leonardi, and Chris Schwiegelshohn. 2019. Algorithms for Fair Team Formation in Online Labour Marketplaces. In *Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, Sihem Amer-Yahia, Mohammad Mahdian, Ashish Goel, Geert-Jan Houben, Kristina Lerman, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.). ACM, 484–490. https://doi.org/10.1145/3308560.3317587

[9] Luca Becchetti, Marc Bury, Vincent Cohen-Addad, Fabrizio Grandoni, and Chris Schwiegelshohn. 2019. Oblivious dimension reduction for *k*-means: beyond subspaces and the Johnson-Lindenstrauss lemma. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*. 1039–1050.

[10] Suman Kalyan Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. 2019. Fair Algorithms for Clustering. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*. 4955–4966.

[11] Ioana O. Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R. Schmidt, and Melanie Schmidt. 2018. On the cost of essentially fair clusterings. *CoRR* abs/1811.10319 (2018). arXiv:1811.10319 http://arxiv.org/abs/1811.10319

[12] Ioana Oriana Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R. Schmidt, and Melanie Schmidt. 2019. On the Cost of Essentially Fair Clusterings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA*. 18:1–18:22.

[13] Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. 2010. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest *k*-subgraph. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*. 201–210. https://doi.org/10.1145/1806689.1806719

[14] Aditya Bhaskara, Moses Charikar, Aravindan Vijayaraghavan, Venkatesan Guruswami, and Yuan Zhou. 2012. Polynomial integrality gaps for strong SDP relaxations of Densest *k*-subgraph. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*. 388–405.

[15] Matteo Böhm, Adriano Fazzone, Stefano Leonardi, Cristina Menghini, and Chris Schwiegelshohn. 2021. Algorithms for fair *k*-clustering with multiple protected attributes. *Oper. Res. Lett.* 49, 5 (2021), 787–789. https://doi.org/10.1016/j.orl.2021.08.011

[16] Matteo Böhm, Adriano Fazzone, Stefano Leonardi, and Chris Schwiegelshohn. 2020. Fair Clustering with Multiple Colors. *CoRR* abs/2002.07892 (2020). arXiv:2002.07892 https://arxiv.org/abs/2002.07892

[17] Avishek Joey Bose and William L. Hamilton. 2019. Compositional Fairness Constraints for Graph Embeddings. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. 715–724.

[18] Vladimir Braverman, Vincent Cohen-Addad, Shaofeng H.-C. Jiang, Robert Krauthgamer, Chris Schwiegelshohn, Mads Bech Toftrup, and Xuan Wu. 2022. The Power of Uniform Sampling for Coresets. In *63rd IEEE Annual Symposium on Foundatons of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*. IEEE, 462–473. https://doi.org/10.1109/FOCS54457.2022.00051

[19] L. Elisa Celis, Lingxiao Huang, and Nisheeth K. Vishnoi. 2018. Multiwinner Voting with Fairness Constraints. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. 144–151.

[20] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. 2018. Ranking with Fairness Constraints. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*. 28:1–28:15.

[21] Moses Charikar. 2000. Greedy approximation algorithms for finding dense components in a graph. In *Approximation Algorithms for Combinatorial Optimization, Third International Workshop, APPROX 2000, Saarbrücken, Germany, September 5-8, 2000, Proceedings*. 84–95. https://doi.org/10.1007/3-540-44436-X_10

[22] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2017. Fair Clustering Through Fairlets. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS)*. 5036–5044.

[23] Miroslav Chlebík and Janka Chlebíková. 2003. Inapproximability Results for Bounded Variants of Optimization Problems. In *Fundamentals of Computation Theory, 14th International Symposium, FCT 2003, Malmö, Sweden, August 12-15, 2003, Proceedings*. 27–38.

[24] Alexandra Chouldechova. 2017. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data* 5, 2 (2017), 153–163.

[25] Hossein Esfandiari, MohammadTaghi Hajiaghayi, and David P. Woodruff. 2016. Brief Announcement: Applications of Uniform Sampling: Densest Subgraph and Beyond. In *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2016, Asilomar State Beach/Pacific Grove, CA, USA, July 11-13, 2016*. 397–399. https://doi.org/10.1145/2935764.2935813

[26] Dan Feldman, Melanie Schmidt, and Christian Sohler. 2013. Turning big data into tiny data: Constant-size coresets for *k*-means, PCA and projective clustering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*. 1434–1453.

[27] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 259–268.

[28] Eugene Fratkin, Brian T Naughton, Douglas L Brutlag, and Serafim Batzoglou. 2006. MotifCut: regulatory motifs finding with maximum density subgraphs. *Bioinformatics* 22, 14 (2006), e150–e157.

[29] David Gibson, Ravi Kumar, and Andrew Tomkins. 2005. Discovering Large Dense Subgraphs in Massive Graphs. In *Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005*. 721–732.

[30] Aristides Gionis, Flavio Junqueira, Vincent Leroy, Marco Serafini, and Ingmar Weber. 2013. Piggybacking on social networks. *Proceedings of the VLDB Endowment* 6, 6 (2013), 409–420.

[31] A. V. Goldberg. 1984. *Finding a Maximum Density Subgraph.* Technical Report UCB/CSD-84-171. EECS Department, University of California, Berkeley.

[32] Teofilo F. Gonzalez. 1985. Clustering to Minimize the Maximum Intercluster Distance. *Theor. Comput. Sci.* 38 (1985), 293–306.

[33] Paula Gordaliza, Eustasio del Barrio, Fabrice Gamboa, and Jean-Michel Loubes. 2019. Obtaining Fairness using Optimal Transport Theory. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. 2357–2365.

[34] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*. 3315–3323.

[35] Lingxiao Huang, Shaofeng H.-C. Jiang, and Nisheeth K. Vishnoi. 2019. Coresets for Clustering with Fairness Constraints. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 7587–7598.

[36] Ravi Kannan and V Vinay. 1999. *Analyzing the structure of large graphs.* Rheinische Friedrich-Wilhelms-Universität Bonn Bonn.

[37] Brent Kitchens, Steven L. Johnson, and Peter Gray. [n. d.]. Understanding Echo Chambers and Filter Bubbles: The Impact of Social Media on Diversification and Partisan Shifts in News Consumption. ([n. d.]).

[38] Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*. 43:1–43:23.

[39] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. 2019. Fair k-Center Clustering for Data Summarization. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. 3448–3457.

[40] Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. 2019. Guarantees for Spectral Clustering with Fairness Constraints. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. 3458–3467.

[41] V. Krebs. [n. d.]. PolBook-Network-Dataset, V. Krebs, unpublished. http://www.orgnet.com/.

[42] H. W. Kuhn and Bryn Yaw. 1955. The Hungarian method for the assignment problem. *Naval Res. Logist. Quart* (1955), 83–97.

[43] Matevž Kunaver and Tomaž Požrl. 2017. Diversity in recommender systems – A survey. *Knowledge-Based Systems* 123 (2017), 154–162. https://doi.org/10.1016/j.knosys.2017.02.009

[44] Yong Liu, Yingtai Xiao, Qiong Wu, Chunyan Miao, Juyong Zhang, Binqiang Zhao, and Haihong Tang. 2020. Diversified Interactive Recommendation with Implicit Feedback. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 04 (2020), 4932–4939. https://doi.org/10.1609/aaai.v34i04.5931

[45] Konstantin Makarychev, Yury Makarychev, and Ilya P. Razenshteyn. 2019. Performance of Johnson-Lindenstrauss transform for *k*-means and *k*-medians clustering. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*. 1027–1038.

[46] Pasin Manurangsi. 2017. Almost-polynomial ratio ETH-hardness of approximating densest k-subgraph. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*. 954–961.

[47] Frank McSherry. 2001. Spectral Partitioning of Random Graphs. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*. 529–537.

[48] Jamie Morgenstern, Samira Samadi, Mohit Singh, Uthaipon Tao Tantipongpipat, and Santosh Vempala. 2019. Fair Dimensionality Reduction and Iterative Rounding for SDPs. *CoRR* abs/1902.11281 (2019).

[49] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 188–197. https://doi.org/10.18653/v1/D19-1018

[50] Alejandro Noriega-Campero, Michiel Bakker, Bernardo Garcia-Bulle, and Alex Pentland. 2018. Active Fairness in Algorithmic Decision Making. *CoRR* abs/1810.00031 (2018). arXiv:1810.00031 http://arxiv.org/abs/1810.00031

[51] Prasad Raghavendra and David Steurer. 2010. Graph expansion and the unique games conjecture. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*. 755–764.

[52] Clemens Rösner and Melanie Schmidt. 2018. Privacy Preserving Clustering with Constraints. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*. 96:1–96:14.

[53] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. 2000. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision* 40, 2 (2000), 99–121.

[54] Samira Samadi, Uthaipon Tao Tantipongpipat, Jamie H. Morgenstern, Mohit Singh, and Santosh Vempala. 2018. The Price of Fair PCA: One Extra dimension. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*. 10999–11010.

[55] Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. 2019. Fair Coresets and Streaming Algorithms for Fair k-means. In *Approximation and Online Algorithms - 17th International Workshop, WAOA 2019, Munich, Germany, September 12-13, 2019, Revised Selected Papers*. 232–251.

[56] Christina Schwind and Jürgen Buder. 2012. Reducing confirmation bias and evaluation bias: When are preference-inconsistent recommendations effective – and when not? *Computers in Human Behavior* 28, 6 (2012), 2280–2290. https://doi.org/10.1016/j.chb.2012.06.035

[57] Uthaipon Tantipongpipat, Samira Samadi, Mohit Singh, Jamie H. Morgenstern, and Santosh S. Vempala. 2019. Multi-Criteria Dimensionality Reduction with Applications to Fairness. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 15135–15145.

[58] Binh Luong Thanh, Salvatore Ruggieri, and Franco Turini. 2011. k-NN as an implementation of situation testing for discrimination discovery and prevention. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 502–510.

[59] Amanda L. Traud, Peter J. Mucha, and Mason A. Porter. 2012. Social structure of Facebook networks. *Physica A* 391, 16 (2012), 4165–4180.

[60] Charalampos E. Tsourakakis, Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Maria A. Tsiarli. 2013. Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*. 104–112.

[61] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. 2017. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification Without Disparate Mistreatment. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*. 1171–1180.

[62] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. 2017. Fairness Constraints: Mechanisms for Fair Classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*. 962–970.

[63] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, Krishna P. Gummadi, and Adrian Weller. 2017. From Parity to Preference-based Notions of Fairness in Classification. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. 228–238.

A APPENDIX

A.1 Proof of Theorem 2.2

Proof. For this proof, we denote by $\hat{\lambda}_1 \geq, \ldots, \geq \hat{\lambda}_n$ the eigenvalues of $(I - ff^T)A(I - ff^T)$ and by $\hat{v}_i$ the ith associated eigenvector. For a vertex i of G_S we denote by $\hat{d}_i$ its degree in G_S. We denote by χ the normalized indicator vector of S and we let $m = |S|$.

As a first step, we summarize some straightforward, yet useful properties of the spectrum of $(I-ff^T)A(I-ff^T)$.

Claim 1. *Whenever $\hat{\lambda}_i \neq 0$ we have:*

$$(I - ff^T)\hat{v}_i = \hat{v}_i \text{ and } \hat{\lambda}_i = \hat{v}_i^T A \hat{v}_i \tag{2}$$

Proof. If $\hat{\lambda}_i \neq 0$, we have: $(I - ff^T)A(I - ff^T)\hat{v}_i = \hat{\lambda}_i\hat{v}_i$. Because $(I - ff^T)$ is a projection matrix, if we pre-multiply both members of the above equation by $(I - ff^T)$ we have $(I - ff^T)A(I - ff^T)\hat{v}_i = \hat{\lambda}_i(I - ff^T)\hat{v}_i$. Subtracting the first equation from the second and recalling that $\hat{\lambda}_i \neq 0$, we immediately obtain the first part of the claim. The second part follows immediately from the first:

$$\hat{\lambda}_i = \hat{v}_i^T(I - ff^T)A(I - ff^T)\hat{v}_i = \hat{v}_i^T A \hat{v}_i.$$

□

It should be noted that, as a consequence of Claim 1, we always have:

$$\hat{\lambda}_1 = \hat{v}_1^T(I - ff^T)A(I - ff^T)\hat{v}_1 = \hat{v}_1^T A \hat{v}_1 \leq v_1^T A v_1 = \lambda_1.$$

Note that this last property does not apply to the other eigenvalues in general. The first important, technical step to prove Theorem 2.2 is showing that the hypothesis $\lambda_1 \geq 4\lambda$ implies that $\hat{\lambda}_2$ cannot be too large. The reason is that, under the assumptions of Theorem 2.2, $\hat{v}_1$ forms a (relatively) small angle with v_1. Because $\hat{v}_2 \perp \hat{v}_1$, this in turn implies that a (relatively) large component of $\hat{v}_2$ belongs to the span of $\{v_2, \ldots, v_n\}$.

Lemma A.1. *Assume the spectrum of A satisfies the condition $\lambda_1 \geq 4\lambda_2$. Then $\hat{\lambda}_2 \leq \frac{3}{4}\lambda_1$.*

Proof. We begin by noting that $f^T\chi = 0$ by definition, which implies that $(I - ff^T)\chi = \chi$. We therefore have:

$$\chi^T(I - ff^T)A(I - ff^T)\chi = \frac{\sum_{i\in S}\hat{d}_i}{\sqrt{|S|}} \geq (1 - \epsilon)d, \tag{3}$$

In the remainder of this proof, we express $\hat{v}_1$ and $\hat{v}_2$ as $\hat{v}_1 = \beta v_1 + q$ and $\hat{v}_2 = \gamma v_1 + z$, where q and z respectively denote $\hat{v}_1$'s and $\hat{v}_2$'s components orthogonal to v_1, the main eigenvector of A. Note that, because v_1, $\hat{v}_1$, and $\hat{v}_2$ have unit norms, we have $\beta^2 + \|q\|^2 = 1$ and $\gamma^2 + \|z\|^2 = 1$. Moreover, by definition of $\hat{\lambda}_1$,[18] from (3) and from the hypotheses of Theorem 2.2 we have:

$$\hat{\lambda}_1 \geq (1 - \epsilon)d \geq (1 - \epsilon)(1 - \theta)d_{\max} > (1 - (\epsilon + \theta))\lambda_1, \tag{4}$$

where the last inequality follows because the main eigenvalue of an adjacency matrix is upper-bounded by the maximum degree of the underlying graph and because $\epsilon\theta > 0$. We further have:

$$\hat{\lambda}_1 = (\beta v_1 + q)^T A(\beta v_1 + q)^T = \beta^2\lambda_1 + q^T A q$$

$$= \beta^2\lambda_1 + \|q\|^2\frac{q^T A q}{\|q\|^2} \leq \beta^2\lambda_1 + (1 - \beta^2)\lambda_2 \leq \beta^2\lambda_1 + \frac{\lambda_1}{4},$$

[18] I.e., $\hat{v}_1$ maximizes the Rayleigh quotient of $(I - ff^T)A(I - ff^T)$.

where the last inequality follows from the hypothesis that $\lambda_1 \geq 4\lambda_2$. Together, the last inequality and (4) imply

$$\beta^2 > \frac{3}{4} - (\epsilon + \theta) \geq \frac{1}{2},$$

from our assumption that $\epsilon + \theta \leq 1/4$. We next show that $\gamma^2 < 1/2$. To this purpose, note that since the $\hat{v}_i$'s are orthonormal, we have $\hat{v}_1^T \hat{v}_2 = 0$, which implies $|\beta\gamma| = |q^T z|$. As a consequence, if $\beta^2 > 1/2$ and $\gamma^2 \geq 1/2$, we would have $|\beta\gamma| > 1/2$, whereas $\beta^2 + \|q\|^2 = 1$ and $\gamma^2 + \|z\|^2 = 1$ would imply $\|q\|^2 < 1/2$ and $\|z\|^2 < 1/2$, whence $|q^T z| \leq \|q\|\|z\| < 1/2$ from the Cauchy–Schwarz inequality, a contradiction.

Next:

$$\begin{aligned}\hat{\lambda}_2 &= (\gamma v_1 + z)^T A (\gamma v_1 + z)^T = \gamma^2 \lambda_1 + z^T A z \\ &\leq \gamma^2 \lambda_1 + \|z\|^2 \lambda_2 \leq \gamma^2 \lambda_1 + \frac{\lambda_1}{4} \leq \frac{3}{4}\lambda_1,\end{aligned}$$

where the first equality follows from Claim 1, the second follows because z is orthogonal to v_1, the third inequality again follows because $z \in \mathbf{span}(v_2, \ldots, v_n)$, the fourth from the lemma's hypothesis and the fifth since we showed above that $\gamma^2 \leq 1/2$. This concludes the proof of Lemma A.1. □

Lemma A.2. *Assume the hypotheses of Theorem 2.2 hold. Then:*

$$\|\chi - \hat{v}_1\|^2 \leq 8(\epsilon + \theta).$$

Proof. We decompose χ along its components respectively parallel and orthogonal to $\hat{v}_1$, namely, $\chi = \alpha \hat{v}_1 + w$, and we note that $\|w\|^2 = 1 - \alpha^2$, as both $\hat{v}_1$ and χ are unit norm vectors. Set $B = (I - ff^T)A(I - ff^T)$ for the sake of space. Moreover, since $\hat{v}_1$ is defined up to its sign, we choose it so that $\alpha \geq 0$. We have:

$$\begin{aligned}\chi^T B \chi &= (\alpha \hat{v}_1 + w)^T B (\alpha \hat{v}_1 + w) = \alpha^2 \hat{\lambda}_1 + w^T B w \\ &\leq \alpha^2 \hat{\lambda}_1 + \hat{\lambda}_2 \|w\|^2 = \alpha^2 \hat{\lambda}_1 + (1 - \alpha^2)\hat{\lambda}_2. \end{aligned} \tag{5}$$

Putting together (3) and (5) yields $\alpha^2 \geq \frac{(1-\epsilon)d - \hat{\lambda}_2}{\hat{\lambda}_1 - \hat{\lambda}_2}$. Now:

$$\begin{aligned}\|\chi - \hat{v}_1\|^2 = (\chi - \hat{v}_1)^T(\chi - \hat{v}_1) = 2 - 2\alpha \leq 2 - 2\alpha^2 \leq 2 - 2\frac{(1-\epsilon)d - \hat{\lambda}_2}{\hat{\lambda}_1 - \hat{\lambda}_2} \leq 2 - 2\frac{(1-\epsilon)(1-\theta)d_{\max} - \hat{\lambda}_2}{\lambda_1 - \hat{\lambda}_2} \\ \leq 2 - 2\frac{(1-\epsilon)(1-\theta)\lambda_1 - \hat{\lambda}_2}{\lambda_1 - \hat{\lambda}_2} < 2 - 2\frac{\lambda_1 - \hat{\lambda}_2 - (\epsilon+\theta)\lambda_1}{\lambda_1 - \hat{\lambda}_2} = 2\frac{(\epsilon+\theta)\lambda_1}{\lambda_1 - \hat{\lambda}_2} \leq 8(\epsilon + \theta).\end{aligned}$$

Here, the third inequality follows from $0 \leq \alpha \leq 1$, the fifth inequality follows from our hypotheses on d and because $\hat{\lambda}_1 \leq \lambda_1$, the sixth inequality again follows because the main eigenvalue of an adjacency matrix is upper-bounded by the maximum degree of the underlying graph, and the last inequality follows from Lemma A.1. □

Corollary A.3. *Under the hypotheses of Lemma A.2, for all but at most* $32m(\epsilon + \theta)$ *vertices in* V *we have: (1)* $\hat{v}_1(i) \geq \frac{1}{2\sqrt{|S|}}$ *if* $i \in S$*, (2)* $\hat{v}_1(i) < \frac{1}{2\sqrt{|S|}}$ *otherwise.*

The algorithm. Our algorithm is based on a sweep of $\hat{v}_1$ [36, 47]. In particular, we run Algorithm GSA (see Algorithm 1) with $M = (I - ff^T)A(I - ff^T)$ and $\Delta = 16(\epsilon + \theta)$.

Corollary A.3 ensures that (1) Algorithm 1 always returns a solution, (2) the solution returned by the algorithm will not be worse than the one obtained by picking i if $\hat{v}_1(i) \geq \frac{1}{2\sqrt{\sqrt{|S|}}}$ and rejecting it otherwise. This concludes the proof of Theorem 2.2. □

A.2 Proof of Theorem 3.4

Proof. We consider the $SSE(\eta, \delta)$ problem. Let $G(V, E, w)$ be a d-regular graph and let $\eta \in (0, 1)$ and $\delta = \delta(\eta) \in (0, 1/2]$ be constants that we will specify later. For any set $S \subset V$ of size $s := \delta \cdot |V|$, we have $w(E_S) := d \cdot s - \Phi(S) \cdot d \cdot s$.

We construct a colored graph $G'(V', E', w')$ by considering all nodes of G to be colored red, and by adding $|V|$ blue nodes. Of these nodes, we select an arbitrary but fixed subset of $\delta \cdot |V|$ blue nodes that we denote by B. Each edge in E_B is weighted uniformly by $t := \frac{2 \cdot d}{s-1}$. The remaining edges are weighted with 0.

Recall that SSEH states that distinguishing between the two cases is NP-hard.

Completeness. If there exists some $S \subset V$ of size s with $\Phi(S) \leq \eta$, then

$$w(E_S) \geq (1 - \eta) \cdot d \cdot s.$$

Then the density of the fair subgraph induced by $S \cup B$ of size $2s$ satisfies

$$D_{S \cup B} = \frac{w(E_S) + w(B)}{2|S|} \geq \frac{(1-\eta) \cdot d \cdot s + t \cdot \binom{s}{2}}{2s} = \frac{(1-\eta) \cdot d + t \cdot \frac{s-1}{2}}{2} \geq (1 - \eta) \cdot d. \tag{6}$$

Soundness. If for all $S \subset V$ of size s, $\Phi(S) \geq 1 - \eta$, then

$$w(E_S) \leq \eta \cdot d \cdot s. \tag{7}$$

Denote the size of the fair densest subgraph C by k. Further, let $C_{red} = C \cap Red$. We will distinguish between four basic cases: (1) $k < 2\mu \cdot s$, (2) $2\mu \cdot s \leq k < 2 \cdot s$, (3) $2 \cdot s \leq k < \frac{2}{\mu} s$, and (4) $\frac{2}{\mu} s \leq k$, where $\mu > 0$ is suitably small constant specified later. We note that cases (1) and (4), and (2) and (3) will turn out to be somewhat symmetric, even if slightly different proofs are required in every case.

First, let $k < 2\mu \cdot s$ and again let B_k be an arbitrary subset of B of size k. Then

$$D_{C_{red} \cup B_k} \leq \frac{d \cdot k + w(B_k)}{2 \cdot k} \leq (1 + 2\mu) \frac{d}{2}, \tag{8}$$

where the first inequality holds due to regularity.

Now, let $2\mu \cdot s \leq k < 2 \cdot s$. We have

$$D_{C_{red} \cup B_k} \leq \frac{\eta \cdot d \cdot s + w(B_k)}{2 \cdot k} \leq \left(1 + \frac{2\eta}{\mu}\right) \frac{d}{2}. \tag{9}$$

Now, let $2 \cdot s \leq k \leq \frac{2}{\mu} \cdot s$. We will first show that

$$w(C) \leq \frac{2}{\mu} \cdot \eta \cdot d \cdot k. \tag{10}$$

For the sake of contradiction, assume that this is not the case. The argument revolves around double counting $w(C)$. There exist $\binom{k}{s}$ subsets of size s of C. Observe that for any such subset S' has weight $w(S') \leq \eta \cdot d \cdot s$ and hence

$$\sum_{S' \subset C \,\wedge\, |S'| = s} w(S') \leq \eta \cdot d \cdot s \cdot \binom{k}{s}.$$

At the same time, every (possibly 0 valued) edge appears in $\binom{k-2}{s-2}$ of these subsets. Hence

$$\sum_{S' \subset C \,\wedge\, |S'| = s} w(S') = w(C) \cdot \binom{k-2}{s-2} > \frac{2}{\mu} \cdot \eta \cdot d \cdot k \cdot \binom{k-2}{s-2}.$$

Combining both equations, we have

$$\frac{2}{\mu}\cdot\eta\cdot d\cdot k\cdot\binom{k-2}{s-2}<\eta\cdot d\cdot s\cdot\binom{k}{s}\Leftrightarrow\frac{2}{\mu}<\frac{k\cdot(k-1)}{s\cdot(s-1)}\frac{s}{k}\leq\frac{2}{\mu},$$

which is a contradiction.

Consider now the density of any fair cut containing $C\cup B_k$, where B_k contains B and $k-s$ further arbitrary blue nodes. We have

$$D_{C_{red}\cup B_k}\quad\leq\quad\frac{\frac{2\eta}{\mu}\cdot d\cdot k+t\cdot\binom{s}{2}}{2\cdot k}\leq\left(1+\frac{2\eta}{\mu}\right)\cdot\frac{d}{2}. \tag{11}$$

Finally, consider the case $k>\frac{2}{\mu}s$. Then the density of any fair cut containing $C\cup B_k$, where B_k contains B and $k-s$ further arbitrary blue nodes, is

$$D_{C_{red}\cup B_k}\quad\leq\quad\frac{d\cdot k+t\cdot\binom{s}{2}}{2\cdot k}\leq(1+2\mu)\frac{d}{2}. \tag{12}$$

We note that bounds from Equations 8 and 11 and Equations 9 and 12 are identical. For $\varepsilon<\frac{1}{4}$, we set $\mu=\frac{\varepsilon}{2}$, $\eta\leq\frac{8}{3}\cdot\varepsilon^2$. Then the ratio between the terms 6 and 8 and the terms 6 and 9 is at least $2-\varepsilon$. Therefore, approximating the fair densest subgraph problem beyond a factor of 2 solves the $SSE(\eta,\delta)$ problem. □

A.3 Proof of Proposition 4.2

Proof. We give a reduction from 3-dimensional matching to fair k-center with three colors; a generalization from ℓ-dimensional matching and ℓ colors is straightforward. Given a hypergraph $G(X\uplus Y\uplus Z,E)$, with disjoint nodes sets X,Y,Z of size n each and k hyperedges $E\subset X\times Y\times Z$, a 3-dimensional matching consists in deciding whether there a exists perfect hypermatching, that is, a collection of n pairwise disjoint hyperedges $H\subseteq E$.

We construct an instance of fair k-center as follows. Each hyperedge $e\in E$ will be mapped to some point p_e and also every node $v\in X,Y,Z$ will be mapped to some point p_v. The points corresponding to hyperedges will be our candidate set of centers C. We now define the distances between our points as follows. For nodes v and hyperedges e, we set

$$d(p_v,p_e)=\begin{cases}1 & \text{if } v\in e\\ 2 & \text{if } v\notin e\end{cases}.$$

The remaining distances are set to 2. This trivially results in a metric.

Now, assume that a perfect hypermatching exists. Then the fair n-center clustering cost is precisely 1. If, however, no perfect hypermatching exists, the cost is 2. Distinguishing between these two cases is NP-hard, hence approximating fair n-center within a factor of $2-\epsilon$ is also NP-hard.

Similarly, if a perfect hypermatching exists, the cost of a fair n-median clustering is precisely $3n$. If the size of the largest hypermatching is $3n-t$, then, in the corresponding fair n-median clustering problem, at least t points have to pay 2, that is, the total cost is at least $3n+t$. Because distinguishing between a perfect hypermatching and a hypermatching of size 95/94 [23] is NP hard, t must be greater than $\frac{1}{95}n$. This in turn implies that it is NP hard to compute an n-fairlet with an approximation factor better than 96/95. . □

A.4 Proof of Proposition 4.4

Proof. We give a reduction from 3-dimensional matching to fair k-center with three colors (a generalization from ℓ-dimensional matching and ℓ colors is straightforward). Given a hypergraph $G(X\uplus Y\uplus Z,E)$, with disjoint nodes sets X,Y,Z of size n each and k hyperedges $E\subset X\times Y\times Z$, 3-dimensional matching consists of deciding whether there exists perfect hypermatching, i.e. a collection of n pairwise disjoint hyperedges $H\subseteq E$.

We construct an instance of fair k-center as follows. Each hyperedge $e \in E$ will be mapped to some point p_e and also every node $v \in X, Y, Z$ will be mapped to some point p_v. The points corresponding to hyperedges will be our candidate set of centers C. We now define the distances between our points as follows. For nodes v and hyperedges e, we set $d(p_v, p_e) = \begin{cases} 1 & \text{if } v \in e \\ 3 & \text{if } v \notin e \end{cases}$. The remaining distances are set to 2. To see that this induces a metric, we only need to observe that the triangle inequality holds for $d(p_v, p_e)$ when $v \notin e$. For any $u \in V \setminus \{v\}$, we have $d(p_v, p_u) + d(p_u, p_e) = 2 + d(p_u, p_e) \geq 3 = d(p_v, p_e)$. For any $e' \in E \setminus \{e\}$, we have $d(p_v, p_{e'}) + d(p_{e'}, p_e) = d(p_v, p_{e'}) + 2 \geq 3 = d(p_v, p_e)$.

Soundness. Let H be a perfect hypermatching. We show that this induces a fair k-center clustering of cost 1. For every hyperedge $e = (x, y, z) \in H$, we set the cluster $C_e := \{p_x, p_y, p_z\}$ with center p_e, the remaining clusters are empty. Clearly, the resulting clustering is fair and the distance of every point to its assigned center is 1.

Completeness. Suppose no perfect hypermatching exists. We show that then there exists no fair clustering of cost 1 (or even less than $3 - \varepsilon$) using the p_e as centers. For the sake of contradiction, suppose there exists such a clustering with clusters $C_1, \ldots C_k$. Any for any cluster with center p_e and more than three nodes must contain a point p_v such that $v \notin e$, and hence have cost 3. Hence all clusters must have size at most 3. The union of these clusters, however, is a perfect hypermatching. □

A.5 Additional Experimental Results for Fair k-Clustering

The following tables contain the average and standard deviation of the cost of the fair-k-median methods across all clustering datasets. Each dataset consists of 8 colors and 100 subsamples.

	$k \in [2, 5]$	$k \in [6, 10]$	$k \in [11, 20]$
Böhm et al. [15]: k=n	4,789.9 *229.5*	4,789.9 *229.5*	4,789.9 *229.5*
k-median++	10,354.9 *1,691.9*	7,218.8 *485.1*	5,710.4 *396.4*
Excellent	10,689.6 *1,541.6*	7,927.3 *431.1*	6,749.5 *343.6*
Böhm et al. [15]	10,855.6 *1,488.8*	8,188.5 *435.3*	7,016.6 *354.3*
Q	10,975.4 *1,499.6*	8,288.9 *449.5*	7,089.5 *367.7*
Algorithm 3	11,082.9 *1,468.3*	8,423.7 *490.0*	7,219.0 *394.0*
Bera et al. [10]	11,148.2 *1,454.4*	8,653.1 *444.2*	7,466.4 *392.0*

Table 5. Average and standard deviation of the cost of the fair-k-median methods on *Diabetes* dataset: 8 colors, 100 subsamples of 1000 distinct points each.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

	$k \in [2,5]$	$k \in [6,10]$	$k \in [11,20]$
Böhm et al. [15]: k=n	15,981,152.5 *1,968,761.8*	15,981,152.5 *1,968,761.8*	15,981,152.5 *1,968,761.8*
k-median++	35,093,729.4 *10,482,181.7*	16,518,716.6 *2,704,863.2*	9,390,692.2 *1,632,344.3*
Excellent	37,334,601.9 *9,646,426.1*	22,164,330.5 *2,448,591.7*	18,136,890.1 *2,030,020.3*
Böhm et al. [15]	38,601,051.2 *9,319,399.3*	23,480,401.1 *2,606,059.3*	18,863,910.7 *2,110,809.0*
Q	39,150,199.6 *9,445,281.9*	23,786,205.0 *2,731,965.7*	19,069,140.1 *2,222,137.3*
Algorithm 3	40,372,180.0 *9,277,749.3*	25,507,540.8 *3,548,581.9*	20,864,471.0 *3,425,165.9*
Bera et al. [10]	38,101,545.1 *9,528,290.7*	22,713,743.0 *2,587,922.8*	18,488,295.4 *2,018,105.9*

Table 6. Average and standard deviation of the cost of the fair-k-median methods on *Adults* dataset: 8 colors, 100 subsamples of 1000 distinct points each.

	$k \in [2,5]$	$k \in [6,10]$	$k \in [11,20]$
Böhm et al. [15]: k=n	72,339,739.1 *3,121,117.5*	72,339,739.1 *3,121,117.5*	72,339,739.1 *3,121,117.5*
k-median++	107,302,571.2 *21,435,714.7*	71,975,930.7 *5,299,320.0*	57,207,897.4 *4,066,524.0*
Excellent	119,312,415.8 *16,746,700.6*	93,053,938.9 *4,189,509.1*	84,871,284.9 *3,599,454.6*
Böhm et al. [15]	122,659,466.9 *16,198,184.0*	96,614,545.3 *4,341,943.3*	87,722,007.7 *3,818,697.4*
Q	124,251,499.5 *16,883,999.8*	97,391,340.5 *4,602,652.1*	88,173,991.3 *3,940,251.9*
Algorithm 3	126,706,176.1 *16,066,799.1*	101,750,820.7 *7,338,227.3*	92,743,080.9 *7,061,486.0*
Bera et al. [10]	123,705,922.5 *17,036,610.1*	97,835,628.9 *4,640,655.8*	89,211,162.0 *3,889,060.7*

Table 7. Average and standard deviation of the cost of the fair-k-median methods on *Credit-Cards* dataset: 8 colors, 100 subsamples of 1000 distinct points each.

	$k \in [2,5]$	$k \in [6,10]$	$k \in [11,20]$
Böhm et al. [15]: k=n	434,892.4 *63,810.1*	434,892.4 *63,810.1*	434,892.4 *63,810.1*
k-median++	699,773.5 *178,243.2*	375,005.8 *53,842.9*	240,197.8 *33,496.6*
Excellent	782,666.7 *148,502.7*	566,505.9 *66,761.8*	501,728.7 *63,811.2*
Böhm et al. [15]	799,852.1 *147,306.8*	581,807.0 *68,002.9*	512,378.1 *64,506.9*
Q	813,091.6 *151,845.3*	585,861.4 *68,185.5*	514,900.4 *64,694.5*
Algorithm 3	840,799.3 *158,718.3*	622,778.7 *89,740.2*	551,295.3 *86,282.3*
Bera et al. [10]	806,863.1 *146,858.0*	595,020.8 *70,544.2*	522,864.6 *65,312.6*

Table 8. Average and standard deviation of the cost of the fair-k-median methods on *Bank* dataset: 8 colors, 100 subsamples of 1000 distinct points each.

	$k \in [2, 5]$	$k \in [6, 10]$	$k \in [11, 20]$
Böhm et al. [15]: k=n	143,647.8 *11,160.9*	143,647.8 *11,160.9*	143,647.8 *11,160.9*
k-median++	95,258.4 *25,778.1*	56,917.8 *5,417.5*	41,405.2 *3,832.5*
Excellent	184,695.5 *20,692.4*	160,658.8 *10,799.8*	153,238.9 *10,856.3*
Böhm et al. [15]	187,589.5 *20,660.7*	163,074.4 *10,881.9*	155,139.9 *10,832.0*
Q	189,288.9 *21,659.0*	163,719.4 *11,099.0*	155,544.5 *10,952.7*
Algorithm 3	205,788.6 *33,060.7*	173,844.1 *14,035.4*	165,075.1 *13,902.7*
Bera et al. [10]	186,506.1 *21,502.2*	162,025.1 *10,575.3*	154,067.1 *10,496.4*

Table 9. Average and standard deviation of the cost of the fair-k-median methods on *CensusII* dataset: 8 colors, 100 subsamples of 1000 distinct points each.

	$k \in [2, 5]$	$k \in [6, 10]$	$k \in [11, 20]$
Böhm et al. [15]: k=n	20,278.4 *280.9*	20,278.4 *280.9*	20,278.4 *280.9*
k-median++	9,970.3 *2,171.1*	6,327.1 *439.9*	5,079.1 *327.3*
Excellent	21,107.9 *300.0*	20,769.6 *276.7*	20,585.8 *273.5*
Böhm et al. [15]	21,299.8 *297.6*	20,948.8 *276.4*	20,728.2 *275.6*
Q	21,340.9 *302.0*	20,965.2 *278.8*	20,736.5 *276.4*
Algorithm 3	21,629.2 *391.5*	21,333.2 *418.7*	21,092.3 *411.6*
Bera et al. [10]	21,959.2 *796.3*	21,098.5 *295.8*	20,793.4 *288.7*

Table 10. Average and standard deviation of the cost of the fair-k-median methods on *Athletes* dataset: 8 colors, 100 subsamples of 1000 distinct points each.

	$k \in [2, 5]$	$k \in [6, 10]$	$k \in [11, 20]$
Böhm et al. [15] + FF: k=n	108,369,855.9 *3,054,639.9*	108,369,855.9 *3,054,639.9*	108,369,855.9 *3,054,639.9*
k-median++	57,863,762.6 *19,309,933.8*	32,275,620.7 *3,110,506.0*	24,170,463.4 *1,923,303.3*
Excellent + FF	171,581,102.9 *8,929,810.2*	174,418,606.9 *6,230,287.9*	171,759,787.4 *5,951,334.0*
Böhm et al. [15] + FF	126,373,453.8 *3,582,880.6*	124,704,090.1 *3,391,359.4*	124,237,402.6 *3,429,534.5*
Q + FF	126,885,878.7 *4,115,222.4*	124,681,188.1 *3,454,367.0*	124,220,881.3 *3,424,855.2*
Algorithm 3 + FF	135,025,164.1 *7,359,895.1*	134,129,796.4 *7,632,188.8*	133,834,734.3 *7,742,148.8*
Bera et al. [10]	—	—	—

Table 11. Average and standard deviation of the cost of the fair-k-median methods combined with fast fairlets decomposition [5] on the *CensusII 450K points* dataset: 8 colors, 100 subsamples of 450,000 distinct points each. The algorithm of Bera et al. [10] was not able to terminate in this dataset.