A Polynomial-time Approximation for Pairwise Fair k-Median Clustering

Sayan Bandyapadhyay
Portland State University

Eden Chlamtáč
Ben Gurion University

Zachary Friggstad
University of Alberta

Mahya Jamshidian
University of Alberta

Yury Makarychev
Toyota Technological Institute at Chicago

Ali Vakilian
Virginia Tech

Abstract

In this work, we study pairwise fair k-Median with $\ell \geq 2$ groups, where for every cluster C and every group $i \in [\ell]$, the number of points in C from group i must be at most t times the number of points in C from any other group $j \in [\ell]$, for a given integer t. Only bicriteria approximation and exponential-time algorithms follow for this problem from the prior work on fair clustering problems when $\ell > 2$.

We present the first polynomial-time $O(k^2 \cdot \ell \cdot t)$-approximation for this problem that does not violate the fairness constraints. We also implemented our algorithm on a variety of datasets to test the "price of fairness" achieved by our approach in real data, which turned out to be significantly smaller than the theoretical guarantee.

1 INTRODUCTION

Clustering is a fundamental task in machine learning and data mining aimed at dividing a set of data items into several groups or clusters, such that each group contains similar data items. Typically, the similarity between data items is measured using a metric distance function. Clustering is often modeled as an optimization problem where the objective is to minimize a global cost function that reflects the quality of the clusters; this function varies depending on the application. Among the many cost functions studied for clustering, the most popular are k-Median [Cohen-Addad et al., 2025], k-Means [Ahmadian et al., 2017], and k-Center [Gonzalez, 1985, Hochbaum and Shmoys, 1986]. These objectives generally aim to minimize the variance within the clusters, serving as a proxy for grouping similar data items.

Motivated by the recent trend in research on fairness in artificial intelligence, clustering problems have been studied extensively with fairness constraints, commonly known as *fair clustering* problems. In this work, we focus on Pairwise Fair Clustering.

Definition 1 (Pairwise Fair Clustering)**.** *We are given a dataset of n points in a metric space (Ω, d), which is partitioned into ℓ disjoint groups $P_1, \ldots, P_\ell$. The input also includes an integer balance parameter $t \geq 1$. A clustering is defined as* pairwise fair *if for any cluster C and any pair of groups P_i and P_j, the number of points from group P_i in C is at least $1/t$ times and at most t times the number of points from group P_j in C. The goal is to find a pairwise fair clustering that minimizes a given cost function.*

The Pairwise Fair Clustering formulation is a natural generalization of the seminal fair clustering model of Chierichetti et al. [2017], extending it to settings with more than two protected groups. Their work initiated an active line of research by formalizing fairness via a single balance parameter, t. Since their work [Chierichetti et al., 2017], other models have also been proposed to handle multiple groups. One notable example is the (α, β)-fairness model, proposed by Bera et al. [2019], Bercea et al. [2019], which requires the fraction of points from each group i in any given cluster to be within a specified range (α_i, β_i). Our Pairwise Fair Clustering model provides a distinct but related approach to this multi-group fairness problem.

While Pairwise Fair Clustering has been studied across various clustering objectives, no polynomial-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

time algorithm with a bounded approximation guarantee was known until very recently. The first such result was provided by Bandyapadhyay et al. [2025] for k-Center, achieving a 33-approximation using a tree-based linear programming (LP) rounding scheme. For k-Median objective, polynomial time approximation is known only in two special cases: (i) the two-group case, i.e., when $\ell = 2$, an $O(t)$-approximation [Chierichetti et al., 2017], and (ii) the fully-balanced case, i.e., when $t = 1$, an $(\alpha + 2)$-approximation [Böhm et al., 2020], where α is the approximation factor for k-Median in the unconstrained case.

Although designing polynomial time "true" approximation algorithms has remained elusive, notable success has been achieved in developing "bi-criteria" constant-approximation algorithms, where the solution is permitted to violate the fairness constraints by an additive term [Bercea et al., 2019, Bera et al., 2019] for a closely related, yet different notion of (α, β)-fair. Such LP-based approaches can be suitably adapted also for Pairwise Fair Clustering. This raises a natural question: *"Is it possible to efficiently convert bi-criteria solutions into true solutions while bounding the approximation guarantee?"*

Our Contributions. In this work, we study Pairwise Fair k-Median with a focus on designing *polynomial time algorithms that do not violate the fairness constraints*. Formally, the goal is to find a pairwise fair clustering $\{X_1, \ldots, X_k\}$ and a set of centers $\{c_1, \ldots, c_k\} \subset \Omega$ that together minimize the k-Median cost, defined as $\sum_{j=1}^{k} \sum_{p \in X_j} d(p, c_j)$.

Our main contribution is a novel technique for converting bi-criteria solutions into *"true"* approximations. Specifically, we design a polynomial time conversion algorithm that yields the *first true approximation* for Pairwise Fair k-Median.

Theorem 1. *There is a polynomial time $O(k^2 \cdot \ell \cdot t)$-approximation for* Pairwise Fair k-Median.

Our algorithm begins by using a constant-factor approximation algorithm for (unconstrained) k-Median to select the set of k centers F. We show that there is an $O(1)$-approximate solution for the Pairwise Fair k-Median instance using centers F, so the focus is then to find a fair assignment to these centers. We use a natural assignment LP relaxation for this task; the main purpose is to find initial (perhaps fractional) values indicating how many clients from each group P_i should be assigned to each center. The LP also helps us decompose the problem into instances where the distance between any two points is $O(k)$ times the optimum solution cost.

We then find an integer assignment of clients that almost matches each of these initial values, with only slight additive violation in the fairness constraint at each center. So far, our assignment is still within a constant-factor of the optimum solution cost. Our final and most important step is to show that we can move some points from their current centers to different centers, satisfying the fairness constraints. Moreover, the number of points moved is independent of n and is only a polynomial function of ℓ, k, and t, hence our approximation guarantee. The details of this algorithm are presented in Section 2 with the proofs of some supporting claims being deferred to Appendix B.

While our approximation guarantee is independent of the number of points, it still may seem a bit large. However, one might expect in practice that the number of points to be clustered is considerably larger than the parameters ℓ, k, and t. In such cases, the additional cost of reassigning a bounded number of points is expected to be small, suggesting that the algorithm will perform much better in practice than its worst-case guarantee indicates.

We empirically evaluated our algorithm on a variety of datasets to investigate this. Our experiments were designed to achieve two goals: first, to test if the algorithm's practical performance is better than its worst-case guarantee, and second, to quantify the "price of fairness" by measuring the increase in cost compared to the initial, unconstrained k-Median. These are presented in Section 3 and Appendix C.

We also provide two hardness of approximation results. First, we show that Pairwise Fair k-Median is as hard as the well-studied Capacitated k-Median, up to a factor of $(1 + \epsilon)$ for any $\epsilon > 0$. Despite extensive research on this problem, the best-known polynomial time approximation factor remains to be $\Omega(\log k)$ [Adamczyk et al., 2019]. Our polynomial time reduction demonstrates that if Pairwise Fair k-Median admits a $o(\log k)$-approximation, so does Capacitated k-Median.

Second, we study the complexity of the variant of Pairwise Fair k-Median when the groups $P_1, \ldots, P_\ell$ are not disjoint, i.e., a point can be included in multiple groups. We remark that the bi-criteria approximation of Bera et al. [2019], where they studied a similar cluster balanced fairness notion (with upper and lower bound parameters), also works for this variant, albeit with a slightly larger additive violation. We prove that Pairwise Fair k-Median with overlapping groups is NP-hard to approximate within a factor of $n^{1-\varepsilon}$ for every $\varepsilon > 0$, even when $k = 2$. An overview of these hardness results is presented in Section 4 with details deferred to Appendix A.

Other Related Work. Constant approximation factors are known for Pairwise Fair Clustering, based on fair coreset construction, with a runtime of $(k\ell)^{O(k)} \cdot n^{O(1)}$ [Bandyapadhyay et al., 2024] (denoted as (t,k)-fair clustering), specifically a $(3+\epsilon)$-approximation for k-Median. When $\ell = 2$, Pairwise Fair k-Center admits an improved 4-approximation [Chierichetti et al., 2017]. The Euclidean version of the problem has also been studied with k-Means and k-Median costs. Schmidt et al. [2019] gave an $n^{O(k/\epsilon)}$ time $(1+\epsilon)$-approximation for two groups. Böhm et al. [2020] obtained an $n^{f(k/\epsilon)}$ time $(1+\epsilon)$-approximation in a restricted setting, with f being a polynomial function. Lastly, Backurs et al. [2019] designed a near-linear time $O(d \cdot \log n)$-approximation when $\ell = 2$ and points are in $\mathbb{R}^d$.

Besides Pairwise Fair Clustering, several other notion of fair clustering have been proposed [Chhabra et al., 2021]. Below, we survey some related to ours.

Rösner and Schmidt [2018] studied the problem where a clustering is called fair if, for every cluster C and every group $1 \le i \le \ell$, the ratio $\frac{|P_i \cap C|}{|C|}$ is equal to the ratio $\frac{|P_i|}{|P|}$. They obtained an LP-based $O(1)$-approximation for this problem with k-Center cost. Bercea et al. [2019] and Bera et al. [2019] studied (α,β)-fair clustering, and designed bi-criteria $O(1)$-approximations for k-Median and k-Means objectives, with additive fairness violation. Schmidt et al. [2019] studied composable coreset construction for this notion. Subsequently, Dai et al. [2022] proposed an $O(\log k)$-approximation for (α,β)-fair k-Median that runs in time $n^{O(\ell)}$. Concurrent to Bercea et al. [2019], Bera et al. [2019], Ahmadian et al. [2019] studied k-Center upper bound requirement only (i.e., $\alpha_i = 0$, $\forall i \in [\ell]$). They devised a bi-criteria approximation with an additive ± 2 violation. Furthermore, Li et al. [2010] considered ℓ-diversity in clustering, in which the points are partitioned into clusters of size at least ℓ, with all points in each cluster belonging to distinct groups. Their formulation does not require the number of clusters to be bounded by k. However, if one additionally enforces a k-clustering, for example by merging the clusters they construct, then no group can make up more than a $1/\ell$ fraction of any resulting cluster. In this sense, their notion can be viewed as a special case of the (α_i,β_i)-fairness framework of [Bera et al., 2019], with $\alpha_i = 0$ and $\beta_i = 1/\ell$ for every group i. This is unrelated to our pairwise fairness notion, which concerns pairwise balance rather than global balance within clusters.

Fair clustering has been studied with various other fairness notions and models, including *fair center representation* [Kleindessner et al., 2019, Chiplunkar et al., 2020, Jones et al., 2020, Hotegni et al., 2023], *proportional fairness* [Chen et al., 2019, Micha and Shah, 2020], *min-max fairness* [Abbasi et al., 2021, Ghadiri et al., 2021, Makarychev and Vakilian, 2021, Chlamtáč et al., 2022, Ghadiri et al., 2022, Gupta et al., 2022], *fair colorful* [Bandyapadhyay et al., 2019, Jia et al., 2020, Anegg et al., 2020], *individual fairness notions* [Jung et al., 2020, Mahabadi and Vakilian, 2020, Negahbani and Chakrabarty, 2021, Brubach et al., 2020, 2021, Ahmadi et al., 2022], and *fair hierarchical* [Ahmadian et al., 2020, Knittel et al., 2023a,b].

2 THE APPROXIMATION ALGORITHM

In this section, we prove Theorem 1. The steps will be described in the following subsections and the entire algorithm will be summarized at the end of this section. We assume $t \ge 2$ as the case $t = 1$ has already been studied and has a constant-factor approximation [Böhm et al., 2020].

We begin by setting up some notation. Let $[\ell] = \{1, 2, \ldots, \ell\}$ and $P = \cup_{a \in [\ell]} P_a$. For any assignment $\sigma : P \to F$ for a subset $F \subset \Omega$, its cost, denoted by $cost(\sigma)$, is $\sum_{v \in P} d(v, \sigma(v))$. Moreover, σ is called *t-balanced* or *fair* if for any $c \in F$ and $a, b \in [\ell]$, $|\sigma^{-1}(c) \cap P_a| \le t \cdot |\sigma^{-1}(c) \cap P_b|$. Note that $\sigma^{-1}(c)$ defines a cluster for each $c \in F$, and so we define the t-balance of the clusters accordingly. Let OPT be the optimal cost for the Pairwise Fair k-Median instance, $F^* = \{c_1^*, \ldots, c_k^*\}$ be a corresponding optimal set of centers, and $\{C_1^*, \ldots, C_k^*\}$ be a corresponding optimal fair clustering.

Observation 1 $(*)$**.** *Consider two t-balanced clusters with b_i and b_i' points, respectively from each group $1 \le i \le \ell$. Then, their union is also t-balanced.*

The proofs marked with $(*)$ are deferred to Appendix. Our algorithm has three major steps described below.

2.1 Step 1: Computation of vanilla k centers

To begin, we use an existing α-approximation for k-Median on the point set P to compute a set of k centers $F \subset \Omega$. Let $\sigma : P \to F$ be the assignment function that maps each point in P to a nearest center in F. Currently, the best polynomial time approximation algorithm for k-Median is a 2.675-approximation [Byrka et al., 2014].

In particular, there exists an $O(1)$-approximation for pairwise fair k-Median using centers F.

Lemma 1 $(*)$**.** *There exists a fair assignment $\gamma : P \to F$ such that $cost(\gamma) \le (2+\alpha) \cdot OPT$.*

However, the more involved component in our algorithm for Pairwise Fair Clustering is to find a fair assignment from P to F.

2.2 Step 2: Finding a low-cost, nearly-fair assignment

We begin by introducing an LP relaxation for Pairwise Fair Clustering using *fixed* centers F. Let $x_{j,i}$ be a variable indicating we assign client $j \in P$ to center $i \in F$.

$$\begin{aligned}
\min \quad & \sum_{i\in F}\sum_{j\in P} d(i,j)\cdot x_{j,i} && [LP(D)]\\
\text{s.t.} \quad & \sum_{i\in F} x_{j,i} = 1, \quad \forall j \in P\\
& \sum_{j\in P_a} x_{j,i} \le t \cdot \sum_{j'\in P_b} x_{j',i}, \quad \forall i \in F, a, b \in [\ell]\\
& x_{j,i} = 0, \quad \text{if } d(i,j) > D\\
& x_{j,i} \ge 0, \quad \forall j \in P, i \in F
\end{aligned}$$

In the rest of the algorithm, we try all distances D of the form $d(i,j)$ for some $i \in F$ and some $j \in P$. We solve linear program $LP(D)$ below for each such distance D, rejecting any for which $LP(D)$ is infeasible. For each value D where the LP is feasible, we run the rest of the algorithm and return the best Pairwise Fair Clustering solution found over all values D for which $LP(D)$ is feasible.

For any D and any feasible solution x to $LP(D)$, let $G(D,x)$ be the *support* graph with vertices $F \cup P$ and edges $\{\{i,j\} : i \in F, j \in P, x_{i,j} > 0\}$.

Lemma 2. *Let γ be the assignment from Lemma 1 and let $D = \max_{j\in P} d(j,\gamma(j))$. Then $LP(D)$ is feasible and has optimal solution value at most $(2+\alpha)\cdot OPT$. For any $j \in P$ and $i \in F$ in the same connected component $C \subseteq F \cup P$ of $G(D,x)$ (where x is any feasible solution to $LP(D)$) we have $d(i,j) \le 2\cdot|F\cap C|\cdot D$.*

Proof. That $LP(D)$ is feasible and has value at most $(2+\alpha)\cdot OPT$ is simply because the natural $\{0,1\}$-assignment corresponding to γ is a feasible solution to the LP.

Now consider any $i \in F$ and $j \in P$ in the same connected component in $G(D,x)$. Let $j = j_1, i_1, j_2, i_2, \ldots, j_b, i_b = i$ be any path in $G(D,x)$ with the minimum number of edge hops. Each hop uses an edge with distance at most D by the LP constraints, so $d(i,j) \le \sum_{a=1}^{b-1}(d(j_a,i_a) + d(i_a,j_{a+1})) + d(j_b,i_b) \le 2\cdot|F\cap C|\cdot D$, where the last bound is because the path is a minimum-hop path, so the points $j_1, i_1, j_2, i_2, \ldots, j_b, i_b$ are distinct. □

From this point on, we will analyze the algorithm for the case $D = \max_{j\in P} d(j,\gamma(j)) \le (2+\alpha)\cdot OPT$. Since our final algorithm takes the best solution over all D, the final solution's cost will be no worse than the cost of the solution in this case. We remark here that our final approximation guarantee will be of the form $O(OPT) + O(k^2\cdot\ell\cdot t\cdot D)$. One would often expect D to be much smaller than $cost(\gamma)$ so the approximation guarantee is likely to be much better in practice.

We now describe a simple procedure that rounds an optimal solution x^* to $LP(D)$ to obtain a low-cost integer assignment of points that is close to a fair assignment.

Observation 2. *The restriction of x^* to any connected component $C \subseteq P \cup F$ of $G(D,x^*)$ is a feasible (fractional) solution to the* Pairwise Fair Clustering *instance with clients $P \cap C$. Furthermore, $P \cap C$ is balanced, meaning $|P_a \cap C| \le t \cdot |P_{a'} \cap C|$ for any two $a, a' \in [\ell]$.*

Proof. To see that $P \cap C$ is balanced, note that the fractional assignment to all centers in $F \cap C$ is balanced. Since the total number of clients in $P_a \cap C$ for each $a \in [\ell]$ is the sum of their fractional assignments to each center in $F \cap C$. So by Observation 1, more precisely a straightforward generalization of it to fractional assignments, $P \cap C$ is balanced. □

From now on, we may assume $G(D,x^*)$ is connected. If it was not, we run the following algorithm on the restriction of x^* to each connected component and output the union of the solutions. For each component C let ν_C be the value of x^* restricted to C. Our algorithm will find a solution with cost only $O(|F\cap C|^2\cdot\ell\cdot t\cdot D)$ more than ν_C in each component C. Adding this for each component C shows the final solution cost would be at most $O(k^2\cdot\ell\cdot t\cdot D) + (2+\alpha)\cdot OPT = O(k^2\cdot\ell\cdot t\cdot OPT)$, as required.

The initial nearly-fair solution is obtained from an optimal solution x^* as follows. First, for each $i \in F$, set $\ell_i = \min_{a\in[\ell]} \sum_{j\in P_a} x^*_{j,i}$. Here, ℓ_i represents the smallest amount that any group of clients is fractionally assigned to center i. Then find a min-cost integral assignment $\sigma_{int} : P \to F$, where $|\sigma_{int}^{-1}(i) \cap P_a| \in [\lfloor l_i \rfloor, \lceil t\cdot l_i \rceil]$.

Lemma 3. *Such assignment σ_{int} exists and has cost at most the optimum solution value of $LP(D)$.*

Proof. Consider a bipartite graph with nodes P on the left side and nodes $\{(i,c) : i \in F, a \in [\ell]\}$ on the right (i.e. a node for each group of clients for each center in F). For each $j \in P$ with, say, $j \in P_a$ and each $i \in F$ we include an edge $\{j,(i,a)\}$ with cost $d(j,i)$. In this way, x^* describes a fractional assignment such

that each $j \in P$ is assigned to an extent of exactly 1 to nodes on the right side and each node (i, a) on the right has received a total assignment between $\lfloor \ell_i \rfloor$ and $\lceil t \cdot \ell_i \rceil$.

Furthermore, the cost of this fractional assignment is exactly the optimal LP solution cost. By integrality of the bipartite assignment polytope with integer upper and lower bounds, there is an integer assignment σ_{int} of no greater cost. Such an assignment can be computed using any polynomial time minimum-cost assignment algorithm. □

As a result, $cost(\sigma_{int})$ is still $O(OPT)$ and the assignment σ_{int} is nearly fair in that for any $i \in F$ and any $a, b \in [\ell]$ we have $|\sigma^{-1}(i) \cap P_a| \leq t \cdot |\sigma^{-1}(i) \cap P_b| + t$. The fairness condition is only violated in extreme cases where we have exactly $\lfloor l_i \rfloor$ clients of some color sent to center i and strictly more than $t \cdot \lfloor l_i \rfloor$ clients of some other color assigned to center i. That is, we have $\lfloor t \cdot \ell_i \rfloor \leq t \cdot \lfloor \ell_i \rfloor + t$, so if i receives at least $\lfloor \ell_i \rfloor + 1$ clients of each color then the assignment to i would indeed be fair.

In the remainder of the algorithm, we change the destination of $O(k \cdot \ell \cdot t)$ clients in the assignment σ_{int} to get a truly fair solution. Each client that has its destination changed will be moved to another center in its connected component, so by Lemma 2 the total cost will increase by at most $O(k^2 \cdot \ell \cdot t \cdot D)$.

2.3 Step 3: Fixing the assignment

To modify σ_{int} to make it a feasible Pairwise Fair Clustering solution, we will first unassign a limited number of clients to get a fair partial assignment and then reassign them to new locations.

Define a set S of **unassigned** clients as follows: for each $i \in F$ and each color a let $\Delta_{i,a} = \max\{0, |\sigma_{int}^{-1}(i) \cap P_a| - t \cdot \lfloor \ell_i \rfloor\}$ denote the number of clients of color a assigned to i in excess of t times the lower bound used in the computation of σ_{int}. If $\Delta_{i,a} > 0$, add any $\Delta_{i,a}$ clients from $\sigma_{int}^{-1}(i) \cap P_a$ to S.

Claim 1. $|S| \leq k \cdot \ell \cdot t$.

Proof. By the upper bound imposed when σ is computed, we have $|\sigma_{int}^{-1}(i) \cap P_a| \leq \lceil t \cdot \ell_i \rceil$. So $\Delta_{i,a} \leq \lceil t \cdot \ell_i \rceil - t \cdot \lfloor \ell_i \rfloor \leq t$. Summing this bound for all $i \in F$ and $a \in [\ell]$ yields claim. □

Observe that if we ignore S, each center i receives between $\lfloor \ell_i \rfloor$ and $t \cdot \lfloor \ell_i \rfloor$ clients of each color, so this partial assignment is fair. Our next task is to reassign all clients in S such that the fairness remains preserved. In doing so, we may reassign a limited number of additional clients beyond those in S.

Let σ be the current partial assignment, initially it is the restriction of σ_{int} to $P \setminus S$, i.e. those clients that were not unassigned. We also maintain integer bounds ℓ'_i for each center, initially set $\ell'_i = \min_{a \in [\ell]} |\sigma^{-1}(i) \cap P_a|$.

Invariant: We will maintain for each center i and each color a that $|\sigma^{-1}(i) \cap P_a| \in [\ell'_i, t \cdot \ell'_i]$, i.e. that the partial assignment is fair. This is true for the initial partial assignment σ by how we unassigned clients and by how we set ℓ'_i.

Let i^* denote a particular center in F, this may be arbitrarily chosen but should remain fixed throughout the algorithm. Repeat following while $S \neq \emptyset$. Intuitively, the following steps will check if some unassigned client can be assigned to any center while preserving fairness of the current assignment. If not, it moves some clients from their currently-assigned center to i^* so that some unassigned client can also be assigned to i^* while maintaining fairness.

- **First Case**: Formally, if there is some $j \in S$ with, say, $j \in P_a$ and some $i \in F$ such that $|\sigma^{-1}(i) \cap P_a| < t \cdot \ell'_i$, set $\sigma(j) := i$ and remove j from S (j is now assigned).
- **Second Case**: Otherwise, pick any $j \in S$ and let $a \in [\ell]$ be such that $j \in P_a$. Observe $|\sigma^{-1}(i^*) \cap P_a| = t \cdot \ell'_{i^*}$ since the previous case did not apply. Let $A = \{b \in [\ell] : |\sigma^{-1}(i^*) \cap P_b| = \ell'_{i^*}\}$. For each $b \in A$, pick any one $i_b \in F$ with $|\sigma^{-1}(i_b) \cap P_b| > \ell'_{i_b}$. Then pick any one $j_b \in P_b$ with $\sigma(j_b) = i_b$ and reassign $\sigma(j_b) := i^*$. After doing so for all $j_b \in A$, set $\sigma(j) := i^*$, remove j from S, and update $\ell'_{i^*} = \ell'_{i^*} + 1$.

In the second step, it could be that $A = \emptyset$: recall the invariant maintains that each center has between ℓ'_i and $t \cdot \ell'_i$ clients from each group but we do not necessarily insist that the "lower bound" ℓ'_i is met. If $A = \emptyset$, then the second step merely increases ℓ'_{i^*} so j can be assigned to i^* while maintaining the invariants.

Next, we prove that there is some $i_b \in F$ for each $b \in A$ in the second case. Given this, it is straightforward to verify the invariants hold after each iteration.

Lemma 4. *Whenever the second case is executed, we can always find a center i_b with $|\sigma^{-1}(i_b) \cap P_b| > \ell'_{i_b}$ for each $b \in A$.*

Proof. Suppose otherwise, i.e. that for some $b \in A$ we have $|\sigma^{-1}(i) \cap P_b| = \ell'_i$ for every $i \in F$. Since the first case does not apply, then every $j \in P_b$ must be assigned to some center as any unassigned $j \in P_b$ could be assigned to *any* center in this case (note we are using $t > 1$ in this argument). So $|P_b| = \sum_{i \in F} \ell'_i$. On the other hand, since there exists a $j \in P_a$ that cannot

be assigned to any center (again since the first case does not apply) and since j itself is not yet assigned, $|P_a| \geq 1 + t \cdot \sum_{i \in F} \ell'_i$. This contradicts the feasibility assumption of the given instance, i.e., $|P_a| \leq t \cdot |P_b|$. □

We now bound the total number of clients j with $\sigma(j) \neq \sigma_{int}(j)$ after the algorithm concludes. For any such j, either $j \in S$ or j was moved in the second case above in order to increase the lower bound ℓ'_{i^*}. Each time this lower bound was increased, at most ℓ clients were reassigned. So a simple bound on the number of clients reassigned this way is $\ell \cdot |S| \leq k \cdot \ell^2 \cdot t$.

But we can refine this analysis a bit by showing the second case only executes at most k times in total. The following claim is the starting point for this argument.

Claim 2. *Consider an iteration where the second case is being executed and let $j \in S$ be as in the description of this case where, say, $j \in P_a$. At this point, no client in P_a that is currently assigned to i^* at the start of this iteration was ever assigned to a different center in any previous iteration.*

Proof. Suppose otherwise, that some $j' \in P_a$ was earlier assigned to some $i' \neq i^*$ and was reassigned to i^* prior to the iteration being considered. In this earlier iteration the second case was being executed and we had $j \in S$ and $|\sigma^{-1}(i^*) \cap P_a| = \ell'_{i^*} < t \cdot \ell'_{i^*}$ (since j' was moved to i^* in this earlier iteration). This contradicts the fact that $j \in S$ in this earlier iteration and that j could be added to i^* (i.e. the first case could have been executed). □

Towards the refined analysis, if ℓ'_{i^*} was never increased then $\sigma(j) \neq \sigma_{int}(j)$ only for $j \in S$. Otherwise, say ℓ'_{i^*} was increased $\Gamma \geq 1$ times.

Lemma 5. $\Gamma \leq k$.

We remark this would prove that the number of clients j_b that had $\sigma(j_b)$ change over all iterations of the second case would is at most $k \cdot \ell$, as $|A| \leq \ell$ in each iteration. So the total number of points j with $\sigma(j) \neq \sigma_{int}(j)$ would then be at most $|S| + k \cdot \ell$ and the final solution cost would then be $cost(\sigma_{int}) + O(k \cdot D) \cdot (|S| + k \cdot \ell) = O(OPT) + O(k^2 \cdot \ell \cdot t \cdot D) = O(k^2 \cdot \ell \cdot t \cdot OPT)$. That is, the proof of Theorem 1 will be complete.

Proof of Lemma 5. Consider the moment that ℓ'_{i^*} was increased the last time (i.e. becomes equal to $\lfloor \ell_{i^*} \rfloor + \Gamma$). At this point, the algorithm was considering some $j \in S$ with, say, $j \in P_a$ to assign to i^*. By Claim 2, every client $j' \in P_a$ currently assigned to i^* at the start of this iteration was either initially assigned to i^* (i.e. had $\sigma_{int}(j') = i^*$ and was not initially unassigned) or was initially in S. By how S was constructed, it initially contained at most $t \cdot k$ clients in P_a so at this point strictly fewer than $t \cdot k$ clients have been added to i^* (recalling $j \in S \cap P_a$ has not yet been added to i^* at this point).

At the start of the algorithm, we had $|\sigma^{-1}(i^*) \cap P_a| \leq t \cdot \lfloor \ell_i \rfloor$ (using the initial assignment σ of $P \setminus S$) and the moment before ℓ'_{i^*} is increased to Γ we have $|\sigma^{-1}(i^*) \cap P_a| = t \cdot (\lfloor \ell_i \rfloor + \Gamma - 1)$. So at least $t \cdot (\Gamma - 1)$ clients in $S \cap P_a$ have been added to i^* since the start of the algorithm. By the previous paragraph, this also at most $t \cdot k - 1$, i.e. $t \cdot (\Gamma - 1) < t \cdot k$. So $\Gamma - 1 < k$, meaning $\Gamma \leq k$ as both values are integers. □

The entire algorithm is summarized in Algorithm 1.

Algorithm 1 PAIRWISE FAIR CLUSTERING

```
Use an O(1)-approximation for k-MEDIAN to get a set of centers F.
CAND ← ∅
for each D ∈ {d(j,i) : j ∈ P, i ∈ F} do
    if LP(D) is infeasible, then break
    let x* be an optimal solution to LP(D)
    let ℓ_i = min_{a∈[ℓ]} Σ_{j∈P_a} x*_{j,i} for each i ∈ F
    for each connected comp. C of G(D, x*) do
        compute a min-cost assignment σ_int : P∩C → F∩C satisfying |σ_int^{-1}(i) ∩ P_a| ∈ [⌊ℓ_i⌋, ⌈t · ℓ_i⌉] for each i ∈ F∩C
        run the fixing procedure in Section 2.3 to get a feasible PAIRWISE FAIR CLUSTERING solution σ for this component C
    end for
    let σ' be the assignment obtained by combining all assignments σ for components of G(D, x*).
    CAND ← CAND ∪ {σ'}
end for
return the cheapest solution in CAND
```

3 EXPERIMENTS

Our empirical evaluation builds on prior work on the fair clustering framework such as in [Bera et al., 2019, Backurs et al., 2019, Chierichetti et al., 2017]. In particular, we use the k-MEDIAN implementation of Bera et al. [2019] to select the centers, then apply our algorithm to find a fair solution. We focus on measuring the "price of fairness", which is the increase in clustering cost compared to the unconstrained clustering solution.

We use the four datasets from the UCI database[1] that

[1] https://archive.ics.uci.edu/datasets

were previously used in this domain; e.g., [Bera et al., 2019, Backurs et al., 2019, Chierichetti et al., 2017]. However, we do not directly compare our final results with theirs since the notion of fairness we are considering differs from theirs.

3.1 Datasets and Setup

We ran our experiments on a Zenbook S14 Windows 11 laptop with Intel Core Ultra 7 CPU, 32 GB RAM, and Pycharm 2025 IDE. We used four datasets, each comprising both numerical and categorical attributes. The datasets and their relevant attributes are summarized below: We used four datasets in our evaluation. The **bank** dataset includes one categorical variable (*marital*) and three numerical variables (*age*, *balance*, and *duration*). The **adult_race** dataset features one categorical variable (*race*) and three numerical variables (*age*, *final-weight*, and *education-num*). Similarly, the **creditcard_education** dataset includes one categorical variable (*education level*) and three numerical variables (*LIMIT_BAL*, *AGE*, and *BILL_AMT1*). Finally, the **census1990_ss** dataset has one categorical variable (*dAge*) and five numerical variables (*dAncstry1*, *dAncstry2*, *iAvail*, and *iCitizen*).

For each dataset, we performed preprocessing to prepare it for clustering. Categorical variables were converted into a one-hot encoded format to ensure they could be treated numerically, while all numerical attributes were standardized to have zero mean and unit variance. This prevents features with larger scales from dominating the distance calculations.

Following the preprocessing steps, we used the standard Euclidean distance for all clustering cost calculations. To manage the computational cost of solving an LP for every possible distance value D, we implemented the geometric grid approach described in Section 3.2. This significantly reduces the number of LPs solved with a negligible impact on solution quality.

3.2 Experimental Methodology

The experiments consist of multiple stages, as below:

1. **Balancing Factor t:** For each dataset, we set the fairness parameter t to be the minimum integer value that for which the input is t-balanced. This imposes the strictest feasible fairness constraint.

2. **Vanilla k-Median Clustering:** We initialize our process by finding cluster centers using a k-Median algorithm, which also serves as our baseline for comparison. Specifically, we employ the 5-approximation single-swap local search algorithm from Bera et al. [2019], with its performance guarantee established by Arya et al. [2004]. The results of this initial clustering are labeled as *vanilla* in our plots.

   Notably, in all our experiments, the vanilla algorithm produced a solution where some demographic groups were not assigned to any center. This outcome implies that the solution by vanilla k-Median is **unboundedly unfair**, corresponding to a fairness parameter of $t = \infty$.

3. **Fairness Adjustment:** One of the bottlenecks in our algorithm is solving $O(n^2)$ linear programs (LPs), one for each possible distance D between points. To make this step more efficient, we solve the LP for only a smaller, geometrically increasing set of values for D. Specifically, if δ is the minimum non-zero distance in the metric space, we test values of the form $\delta \cdot (1.1)^j$ for $j \geq 0$ up to the maximum distance.

   This discretization significantly reduces the number of LPs that need to be solved. While this approach can worsen the theoretical approximation guarantee by a factor of at most 1.1, we expect a minimal impact on the solution quality in practice. We use Gurobi[2] to solve the LPs in our implementation. Furthermore, this step is highly parallelizable, as the LPs for different values of D can be solved independently.

4. **Finding best assignment using another integer program:** Finally, after our Pairwise Fair k-Median algorithm determines the number of clients from each group to be assigned to each center in F, we perform a post-processing step to potentially improve the solution. This step consists of finding the minimum-cost assignment of clients to centers that satisfies these per-center quotas.

   Although this post-processing does not improve the theoretical worst-case approximation guarantee, we observed that it consistently reduces the cost of the final solution in our experiments.

3.3 Results and Analysis

We conducted experiments on datasets with varying cluster counts, k. The results demonstrate how fairness adjustments influence clustering costs. Representative results are presented in Figure 1, where we compare the vanilla and fair clustering results for each dataset. Additional plots can be found in Appendix C, where the difference can be seen to be more pronounced as the number of clusters k increases, which can be expected as it will be harder to create balanced

[2] https://www.gurobi.com

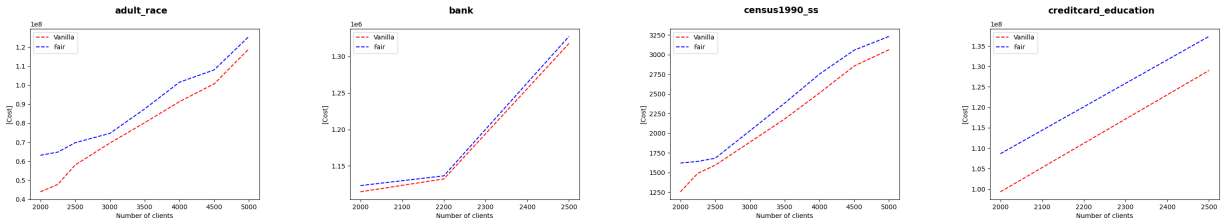

Figure 1: Cost comparison across multiple datasets for $k = 5$. Each point represents the average cost over 5 subsamples of size n, comparing our pairwise-fair k-MEDIAN to the unconstrained k-MEDIAN baseline. The top dashed line in all figures in the fair cost and the bottom one is the vanilla cost.

clusters for larger k. Still, the difference in costs is much better than the worst-case approximation guarantee we produced.

As shown in the runtime plots (Appendix C), the total execution time of our algorithm is dominated by the initial vanilla k-MEDIAN step. The subsequent phases, including solving the LPs and the final client assignment, contribute only marginal overhead. This demonstrates that our method achieves provable fairness with high computational efficiency, adding little cost to a standard clustering pipeline.

In what follows, we have an example of metadata output of an experiment run, including the ratio of the costs as well as the number of points needed to reassign in the fixing step:

Metric	Value
unconstrained k-MEDIAN cost	236,352.57
our fair k-MEDIAN cost	307,917.24
price of fairness	1.30
number of D values tried	113
total runtime of our algorithm	11.03s
number of points to reassign	17

Table 1: Performance on `bank` dataset ($n = 500$, $k = 5$).

Our experiments provides a systematic trade-off between clustering cost, the fairness parameters t, and k. As the fairness parameter t increases; i.e., the constraint is relaxed, the *price of fairness* (gap between the fair and unconstrained k-MEDIAN cost) decreases consistently. In contrast, increasing k tends to *increase* this price of fairness: with more (and thus smaller) clusters, satisfying pairwise-balance constraints becomes harder without incurring additional cost, which widens the gap. Full experimental details and their plots appear in Appendix C.

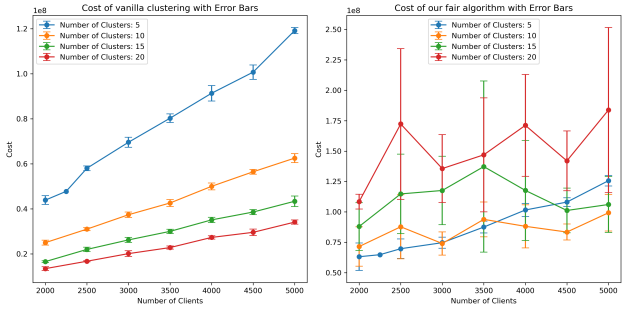

Figure 2: k-MEDIAN cost of the unconstrained and fair algorithms versus k. Each point is the average of 5 trials, with error bars indicating the standard deviation. In the left graph, the lines represent cluster count of 5, 10, 15, and 20, top to bottom; while in the right graph, at the left most point, the lines represent 20, 15, 10, and 5 cluster count, top to bottom.

4 HARDNESS OF APPROXIMATION RESULTS

We provide an overview of our hardness of approximation results for PAIRWISE FAIR k-MEDIAN, and proofs are deferred to Appendix A.

Disjoint Groups. We show that PAIRWISE FAIR k-MEDIAN with disjoint groups is almost as hard as SOFT UNIFORM CAPACITATED k-MEDIAN (CKM). This reduction is significant because, despite extensive research, the best-known polynomial-time approximation for CKM remains $O(\log k)$ [Adamczyk et al., 2019]. Our result demonstrates that a $o(\log k)$-approximation for PAIRWISE FAIR k-MEDIAN would directly translate to an equivalent improvement for CKM. This shows similar hardness for the notion considered by Bera et al. [2019], Bercea et al. [2019] too.

Definition 2 (SOFT UNIFORM CKM)**.** *We are given metric space (X, d) and parameters k and u. Then, the goal is to open k facilities $f_1, \ldots, f_k \in X$ and assign each client (point) $x \in X$ to one of the facilities $f_{a(x)}$ so that at most u clients are assigned to each facility.*

The objective is to minimize $\sum_{x \in X} d(x, f_{a(x)})$.

Theorem 2 (∗)**.** *Suppose there is a polynomial time* α*-approximation algorithm for* Pairwise Fair k-Median *with* $\ell = 2$*. Then* Soft CkM *admits an* $(1+\varepsilon)\alpha$ *approximation for every constant* $\varepsilon > 0$.

Here we sketch the proof of this theorem. Consider an instance (X, d) with k and capacity u of CkM. Using standard rescaling and truncating steps, we may assume that the smallest positive distance in X is at least 1 and the diameter $\mathrm{diam}(X)$ of X is at most polynomial in $n = |X|$. We transform this instance to an instance $\mathcal{I}$ of Pairwise Fair k-Median. The set of locations in $\mathcal{I}$ is $X \cup \{R\}$, where R is an extra point at distance $\mathrm{diam}(X)$ from all other locations $x \in X$. There will be multiple clients/points at every location, as we will describe below. Instance $\mathcal{I}$ has two groups, black G_b and red G_r. Define $W = \lceil k \cdot \mathrm{diam}(X)/\varepsilon \rceil$. We put W black points at every location $x \in X$. We put k red points at R. The parameter t equals uW.

To relate the cost of the two instances, we prove that if the capacitated clustering cost is z, then the cost of $\mathcal{I}$ is at most $Wz + k \cdot \mathrm{diam}(X)$, and if the cost of $\mathcal{I}$ is y, the capacitated clustering cost is at most y/W.

Overlapping Groups. Now, we prove a very strong hardness result for the variant of Pairwise Fair k-Median with overlapping groups, where each point may belong to more than one group. We show that the problem does not admit an $n^{1-\varepsilon}$ approximation for every $\varepsilon > 0$ if P $\neq$ NP, even when $k = 2$.

Theorem 3 (∗)**.** *It is* NP*-hard to approximate* Pairwise Fair k-Median *with overlapping groups within* $(n^{1-\varepsilon})$*-factor for every constant* $\varepsilon > 0$*, even for* $k = 2$.

To prove this hardness result, we present a reduction from the problem of 2-coloring a 3-uniform hypergraph, which is known to be NP-hard [Dinur et al., 2005]. In this problem, given a 3-uniform hypergraph $H = (V, E)$ with $N = |V|$ vertices, the objective is to decide whether there is a coloring of V with two colors such that no hyperedge in E is monochromatic. The full reduction and hardness proof is deferred to Appendix A.2.

Acknowledgements

SB was supported by National Science Foundation Grant 2311397. ZF was supported by an NSERC Discovery Grant. YM was supported by NSF awards CCF-1955173 and ECCS-2216899. The work was done while EC was visiting and supported by TTIC.

References

Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. Fair clustering via equitable group representations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAccT)*, page 504–514, 2021.

Marek Adamczyk, Jaroslaw Byrka, Jan Marcinkowski, Syed Mohammad Meesum, and Michal Wlodarczyk. Constant-factor FPT approximation for capacitated k-median. In *Annual European Symposium on Algorithms (ESA)*, volume 144 of *LIPIcs*, pages 1:1–1:14, 2019.

Saba Ahmadi, Pranjal Awasthi, Samir Khuller, Matthäus Kleindessner, Jamie Morgenstern, Pattara Sukprasert, and Ali Vakilian. Individual preference stability for clustering. In *International Conference on Machine Learning (ICML)*, 2022.

Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 61–72. IEEE Computer Society, 2017.

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

Sara Ahmadian, Alessandro Epasto, Marina Knittel, Ravi Kumar, Mohammad Mahdian, Benjamin Moseley, Philip Pham, Sergei Vassilvitskii, and Yuyan Wang. Fair hierarchical clustering. *Advances in Neural Information Processing Systems*, 33:21050–21060, 2020.

Georg Anegg, Haris Angelidakis, Adam Kurpisz, and Rico Zenklusen. A technique for obtaining true approximations for k-center with covering constraints. In *International conference on integer programming and combinatorial optimization*, pages 52–65. Springer, 2020.

Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3): 544–562, 2004.

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

Sayan Bandyapadhyay, Tanmay Inamdar, Shreyas Pai, and Kasturi Varadarajan. A constant approxima-

tion for colorful k-center. In *27th Annual European Symposium on Algorithms (ESA 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

Sayan Bandyapadhyay, Fedor V Fomin, and Kirill Simonov. On coresets for fair clustering in metric and euclidean spaces and their applications. *Journal of Computer and System Sciences*, 142:103506, 2024.

Sayan Bandyapadhyay, Tianzhi Chen, Zachary Friggstad, and Mahya Jamshidian. A constant-factor approximation for pairwise fair k-center clustering. In Nicole Megow and Amitabh Basu, editors, *Integer Programming and Combinatorial Optimization - 26th International Conference, IPCO 2025, Baltimore, MD, USA, June 11-13, 2025, Proceedings*, volume 15620 of *Lecture Notes in Computer Science*, pages 43–57. Springer, 2025.

Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. In *Advances in Neural Information Processing Systems*, pages 4954–4965, 2019.

Ioana O Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R Schmidt, and Melanie Schmidt. On the cost of essentially fair clusterings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2019.

Matteo Böhm, Adriano Fazzone, Stefano Leonardi, and Chris Schwiegelshohn. Fair clustering with multiple colors. *arXiv preprint arXiv:2002.07892*, 2020.

B Brubach, D Chakrabarti, J Dickerson, A Srinivasan, and L Tsepenekas. Fairness, semi-supervised learning, and more: A general framework for clustering with stochastic pairwise constraints. In *Proc. Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Brian Brubach, Darshan Chakrabarti, John Dickerson, Samir Khuller, Aravind Srinivasan, and Leonidas Tsepenekas. A pairwise fair and community-preserving approach to k-center clustering. In *International Conference on Machine Learning (ICML)*, pages 1178–1189, 2020.

Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 737–756. SIAM, 2014.

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

Anshuman Chhabra, Karina Masalkovaitė, and Prasant Mohapatra. An overview of fairness in clustering. *IEEE Access*, 2021.

Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, pages 5029–5037, 2017.

Ashish Chiplunkar, Sagar Kale, and Sivaramakrishnan Natarajan Ramamoorthy. How to solve fair k-center in massive data models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1877–1886, 2020.

Eden Chlamtáč, Yury Makarychev, and Ali Vakilian. Approximating fair clustering with cascaded norm objectives. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2664–2683, 2022.

Vincent Cohen-Addad, Fabrizio Grandoni, Euiwoong Lee, Chris Schwiegelshohn, and Ola Svensson. A $(2+\varepsilon)$-approximation algorithm for metric k-median. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 615–624, 2025.

Zhen Dai, Yury Makarychev, and Ali Vakilian. Fair representation clustering with several protected classes. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 814–823, 2022.

Irit Dinur, Oded Regev, and Clifford Smyth. The hardness of 3-uniform hypergraph coloring. *Combinatorica*, 25(5):519–535, 2005.

Mehrdad Ghadiri, Samira Samadi, and Santosh S. Vempala. Socially fair k-means clustering. In Madeleine Clare Elish, William Isaac, and Richard S. Zemel, editors, *Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 438–448, 2021.

Mehrdad Ghadiri, Mohit Singh, and Santosh S Vempala. Constant-factor approximation algorithms for socially fair k-clustering. *arXiv preprint arXiv:2206.11210*, 2022.

Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

Swati Gupta, Jai Moondra, and Mohit Singh. Which l_p norm is the fairest? approximations for fair facility location across all "p", 2022.

Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *J. ACM*, 33(3):533–550, 1986. doi: 10.1145/5925.5933. URL https://doi.org/10.1145/5925.5933.

Sedjro Salomon Hotegni, Sepideh Mahabadi, and Ali Vakilian. Approximation algorithms for fair range clustering. In *International Conference on Machine Learning*, pages 13270–13284. PMLR, 2023.

Xinrui Jia, Kshiteej Sheth, and Ola Svensson. Fair colorful k-center clustering. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 209–222. Springer, 2020.

Matthew Jones, Huy Nguyen, and Thy Nguyen. Fair k-centers via maximum matching. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4940–4949, 2020.

Christopher Jung, Sampath Kannan, and Neil Lutz. A center in your neighborhood: Fairness in facility location. In *Proceedings of the Symposium on Foundations of Responsible Computing (FORC)*, page 5:1–5:15, 2020.

Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In *36th International Conference on Machine Learning, ICML 2019*, pages 5984–6003. International Machine Learning Society (IMLS), 2019.

Marina Knittel, Max Springer, John Dickerson, and MohammadTaghi Hajiaghayi. Fair, polylog-approximate low-cost hierarchical clustering. *Advances in Neural Information Processing Systems*, 36:44836–44846, 2023a.

Marina Knittel, Max Springer, John P Dickerson, and MohammadTaghi Hajiaghayi. Generalized reductions: making any hierarchical clustering fair and balanced with low cost. In *International Conference on Machine Learning*, pages 17218–17242. PMLR, 2023b.

Jian Li, Ke Yi, and Qin Zhang. Clustering with diversity. In *International Colloquium on Automata, Languages, and Programming*, pages 188–200. Springer, 2010.

Shi Li. On uniform capacitated k-median beyond the natural lp relaxation. *ACM Transactions on Algorithms (TALG)*, 13(2):1–18, 2017.

Sepideh Mahabadi and Ali Vakilian. Individual fairness for k-clustering. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 6586–6596, 2020.

Yury Makarychev and Ali Vakilian. Approximation algorithms for socially fair clustering. In *Conference on Learning Theory (COLT)*, pages 3246–3264. PMLR, 2021.

Evi Micha and Nisarg Shah. Proportionally fair clustering revisited. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, 2020.

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

Clemens Rösner and Melanie Schmidt. Privacy preserving clustering with constraints. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, 2018.

Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. Fair coresets and streaming algorithms for fair k-means. In *International Workshop on Approximation and Online Algorithms*, pages 232–251, 2019.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] Due to space constraints some proofs are deferred to appendix.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A HARDNESS OF APPROXIMATION RESULTS

In this section, we provide our hardness of approximation results for Pairwise Fair k-Median.

A.1 With Disjoint Groups

We show that Pairwise Fair k-Median with disjoint groups is almost as hard as Soft Uniform Capacitated k-Median (CkM).

Definition 2 (Soft Uniform CkM)**.** *We are given metric space (X, d) and parameters k and u. Then, the goal is to open k facilities $f_1, \ldots, f_k \in X$ and assign each client (point) $x \in X$ to one of the facilities $f_{a(x)}$ so that at most u clients are assigned to each facility. The objective is to minimize $\sum_{x \in X} d(x, f_{a(x)})$.*

There is a more general version of this problem, where the sets of client and facility locations may be different. However, Li [2017] proved that both variants of the problem admit the same approximation. Further, one can consider a hard variant of CkM, in which all opened facilities must be at different locations. Li [2017] showed that if one variant admits an α approximation then the other admits an $O(\alpha)$ approximation.

Theorem 2 $(*)$**.** *Suppose there is a polynomial time α-approximation algorithm for* Pairwise Fair k-Median *with $\ell = 2$. Then* Soft CkM *admits an $(1+\varepsilon)\alpha$ approximation for every constant $\varepsilon > 0$.*

Proof. Consider an instance (X, d) with k and u of CkM. Using standard rescaling and truncating steps, we may assume that the smallest positive distance in X is at least 1 and the diameter $\mathrm{diam}(X)$ of X is at most polynomial in $n = |X|$.

We transform this instance to an instance $\mathcal{I}$ of Pairwise Fair k-Median. The set of locations in $\mathcal{I}$ is $X \cup \{R\}$, where R is an extra point at distance $\mathrm{diam}(X)$ from all other locations $x \in X$. There will be multiple clients/points at every location, as we will describe below. Instance $\mathcal{I}$ has two groups, black G_b and red G_r. Define $W = \lceil k \cdot \mathrm{diam}(X)/\varepsilon \rceil$. We put W black points at every location $x \in X$. We put k red points at R. The parameter t equals uW.

Now we relate the costs of the CkM and Pairwise Fair k-Median instances. Consider a solution for CkM with facilities $(f_1, \ldots, f_k)$ and assignment a. Denote its cost by z. Assume Wlog that at least one client is assigned to every opened facility. We show that there is a corresponding solution for the Pairwise Fair k-Median instance of cost $y \le Wz + k \cdot \mathrm{diam}(X)$. We simply assign every black point at location $x \in X$ to center $f_{a(x)}$. We assign red point number i at R to facility f_i. We open at most k centers. We verify that the fairness constraint holds. Consider a location v with at least one facility. Denote the opened facilities at this location by $f_{i_1}, \ldots, f_{i_q}$. Then, we assign q red points to v and $\sum_{j=1}^{q} W|\{x : a(x) = i_j\}| \le qWu$ black points to location f. The assignment cost is Wz for black points and is at most $k \cdot \mathrm{diam}(X)$ for red points.

Now consider a solution for Pairwise Fair k-Median. We may assume that this solution does not assign any black points to R. If it does, we can reassign these black points and an appropriate number of red points to locations in X without increasing the cost, since the assignment cost for black points may only go down (here we use that $d(R, x) = \mathrm{diam}(X)$ for all $x \in X$).

We treat red points assigned to locations in X as facilities. That is, if red point number i is assigned to x, we open facility f_i at point x in our solution for CkM. We use the assignment of black points to red points, to define an assignment flow from locations in X to facilities f_i such that (i) W units of flow leaves every $x \in X$ and (ii) at most $t = uW$ units of flow enters every facility f_i. We divide all flow amounts by W and get a rescaled flow Φ such that (i′) 1 unit of flow leaves every x and (ii′) at most u units of flow enter every f_i. The cost of this flow with respect to edge costs $d(u, v)$ is at most y/W. Note that if Φ is integral, then it defines an assignment of points in X to facilities in such a way that at most u points are assigned to each facility. Further, the cost of the assignment is $z \le y/W$, as required. So if Φ is integral, we are done. We now show that there is always an *integral* flow Φ' satisfying properties (i′) and (ii′) and of cost at most that of Φ. To this end, we construct an s-t flow network. The network consists of 4 layers. The first layer consists of a single vertex s, the second layer consists of X, the third one consists of $f_1, \ldots, f_k$, and the fourth one of a single vertex t. There is an edge of capacity 1 from source s to each vertex x in X; there is an edge of infinite capacity from each $x \in X$ to each f_i; there is an edge of capacity u from each f_i to t. Each edge from x to f_i has cost $d(x, f_i)$; all other edges are free.

Note that Φ defines a flow from X to $\{f_1, \ldots, f_k\}$, which can be uniquely extended to a feasible flow from s to t. Its cost is $z \leq y/W$ and it saturates all edges from s. Since all the capacities are integral, there exists a feasible integral flow Φ' of cost at most $z \leq y/W$ that also saturates all edges from s. Φ' defines an integral assignment of points in X to facilities $f_1, \ldots, f_k$, as desired.

Now to solve an instance of CkM, we first reduce it to an instance of Pairwise Fair k-Median, run the provided α-approximation algorithm, and then transform the obtained solution to a solution of CkM. Instance $\mathcal{I}$ of Pairwise Fair k-Median has a solution of cost at most $W \cdot OPT + k \cdot \mathrm{diam}(X)$, where OPT is the optimal cost of CkM. Thus, the obtained solution for CkM has cost at most

$$\frac{\alpha(W \cdot OPT + k \cdot \mathrm{diam}(X))}{W} \leq (1+\varepsilon)\alpha \cdot OPT.$$

□

A.2 With Overlapping Groups

Now, we prove a very strong hardness result for the variant of Pairwise Fair k-Median with overlapping groups, where each point may belong to more than one group. We show that the problem does not admit an $n^{1-\varepsilon}$ approximation for every $\varepsilon > 0$ if P $\neq$ NP, even when $k = 2$.

Theorem 3 $(*)$**.** *It is* NP*-hard to approximate* Pairwise Fair k-Median *with overlapping groups within* $(n^{1-\varepsilon})$*-factor for every constant* $\varepsilon > 0$*, even for* $k = 2$*.*

Proof. To prove this hardness result, we will present a reduction from the problem of 2-coloring a 3-uniform hypergraph, which is known to be NP-hard [Dinur et al., 2005]. In this problem, given a 3-uniform hypergraph $H = (V, E)$ with $N = |V|$ vertices, the objective is to decide whether there is a coloring of V in two colors such that no hyperedge in E is monochromatic.

Consider a 3-uniform hypergraph $H = (V, E)$. We define an instance of Pairwise Fair k-Median with overlapping groups as follows:

- The instance has three locations p_1, q, p_2, equipped with the line distance: $d(p_1, q) = d(p_2, q) = 1$, and $d(p_1, p_2) = 2$.
- For every vertex $u \in V$, there is a corresponding point u at location q, which we will identify with u.
- There are N^ρ points at each of the locations p_1 and p_2, where $\rho = 2/\varepsilon$.
- For each hyperedge $e = (v_1, v_2, v_3) \in E$, there is a corresponding group $G_e = \{v_1, v_2, v_3\}$ in the instance. Additionally, there is a group G_0 consisting of all the vertices at locations p_1 and p_2.
- Let $n = 2N^\rho + N$ be the total number of points. Define the fairness parameter t as $t = n$.

Note that the distances between points at the same location are 0. If desired, we can make them strictly positive by making all of them equal to a sufficiently small $0 < \delta \ll 1/N^\rho$. This change will not affect the proof below.

For the chosen value of t, the fairness constraint simply requires that every non-empty cluster must contain at least one point from each group. Consider the following two possibilities.

Hypergraph H is 2-colorable. Let V_1 and V_2 be a valid 2-coloring of V. It defines a clustering of the points: C_1 consists of all the points at location p_1 and all the points (corresponding to the vertices) in V_1; similarly, C_2 consists of all the points at location p_2 and all the points (corresponding to the vertices) in V_2. We open centers for C_1 and C_2 at locations p_1 and p_2, respectively. Since each hyperedge e has at least one vertex from V_1 and one from V_2, both clusters C_1 and C_2 contain at least one representative from each group G_e. Clearly, C_1 and C_2 also contain representatives from G_0. Therefore, clustering C_1, C_2 satisfies the fairness constraint. Its cost is N, since each vertex at location q contributes 1 and all other points contribute 0 to the cost.

Hypergraph H is not 2-colorable. In this case, there is no clustering with two non-empty clusters, satisfying the fairness constraint. If there were such a clustering C_1, C_2 then its restriction to points in V would define a valid 2-coloring of H. Therefore, every fair clustering with $k = 2$ clusters has only one non-empty cluster; that is, all points belong to the same cluster. It is immediate that the cost of the clustering is at least $2N^\rho$.

Now, it is NP-hard to distinguish between the possibilities listed above, and accordingly between the cases when the clustering cost is at most N and is at least $2N^\rho$. We conclude that the problem is NP-hard to approximate within $2N^{\rho-1} = \frac{2N^\rho}{N} \geq \frac{n/2}{n^{\varepsilon/2}} \geq n^{1-\varepsilon}$. □

B MISSING PROOFS

Proof of Observation 1. Consider any two indexes i and j. The union of the two clusters has $b_i + b'_i$ and $b_j + b'_j$ points, respectively from groups i and j. Now, $b_i + b'_i \leq t \cdot b_j + t \cdot b'_j = t \cdot (b_j + b'_j)$. Hence, the merged cluster is also t-balanced. □

Proof of Lemma 1. Let $\phi : F^* \to F$ map each $i \in F^*$ to its nearest $i' \in F$. Consider the following assignment $\gamma : P \to F$. For each $1 \leq i \leq k$, assign all the points in the i-th optimal fair cluster C_i^* to $\phi(c_i^*)$.

First, we prove that γ is fair, i.e., for any $c \in F$ and $i, j \in [\ell]$, $|\gamma^{-1}(c) \cap P_i| \leq t \cdot |\gamma^{-1}(c) \cap P_j|$. Let $C^*_{i^1}, \ldots, C^*_{i^\kappa}$ be the clusters whose points are assigned to c through γ. Then, the set of points that are assigned to c is the union of these clusters. By applying Observation 1 in an inductive manner, we obtain $|\gamma^{-1}(c) \cap P_i| \leq t \cdot |\gamma^{-1}(c) \cap P_j|$.

Next, we bound the cost of γ. Consider any point $p \in P$ and let $p \in C_i^*$.

$$
\begin{aligned}
d(p, \gamma(p)) &= d(p, \phi(c_i^*)) \\
&\leq d(p, c_i^*) + d(c_i^*, \phi(c_i^*)) && \text{(triangle inequality)} \\
&\leq d(p, c_i^*) + d(c_i^*, \sigma(p)) && (\phi(c_i^*) \text{ is the nearest neighbor of } c_i^* \text{ in } F) \\
&\leq d(p, c_i^*) + d(c_i^*, p) + d(p, \sigma(p)) && \text{(triangle inequality)} \\
&= 2d(p, c_i^*) + d(p, \sigma(p))
\end{aligned}
$$

So, $cost(\gamma) \leq \sum_{p \in P} (2d(p, c_i^*) + d(p, \sigma(p))) \leq (2 + \alpha) \cdot OPT$. □

C PLOTS OF RESULTS

Additional plots comparing the vanilla k-MEDIAN clustering cost and the cost of the solutions found by our algorithm are found in this section. While the costs can sometimes be notably more expensive, the approximation guarantee is certainly better than $\Omega(k^2 \cdot \ell \cdot t)$ even if we use the vanilla clustering optimum solution value as a lower bound.

Plots of the running time of vanilla clustering versus our algorithm are also presented. In all cases, it is clear that the bottleneck is the vanilla clustering routine, and the overall algorithm could be sped up by using a faster initial k-MEDIAN approximation.

For each unique number of clients and number of clusters and for each dataset, we ran 10 trials and show the mean of the vanilla k-median cost vs our fair algorithm cost in the figures below. We also plot the clustering time for both the k-median algorithm and out fair algorithm. The results denote how the most time-consuming part of the fair algorithm remains the initial k-median procedure. We also include the error bar plots for all datasets in each category of fair and unfair, each indicating the mean and standard deviation. At last, we analyze the change in the cost as t increases, which shows a mostly downward trend in the cost; which is expected as the fairness parameter is larger, allowing for more imbalance in each cluster.

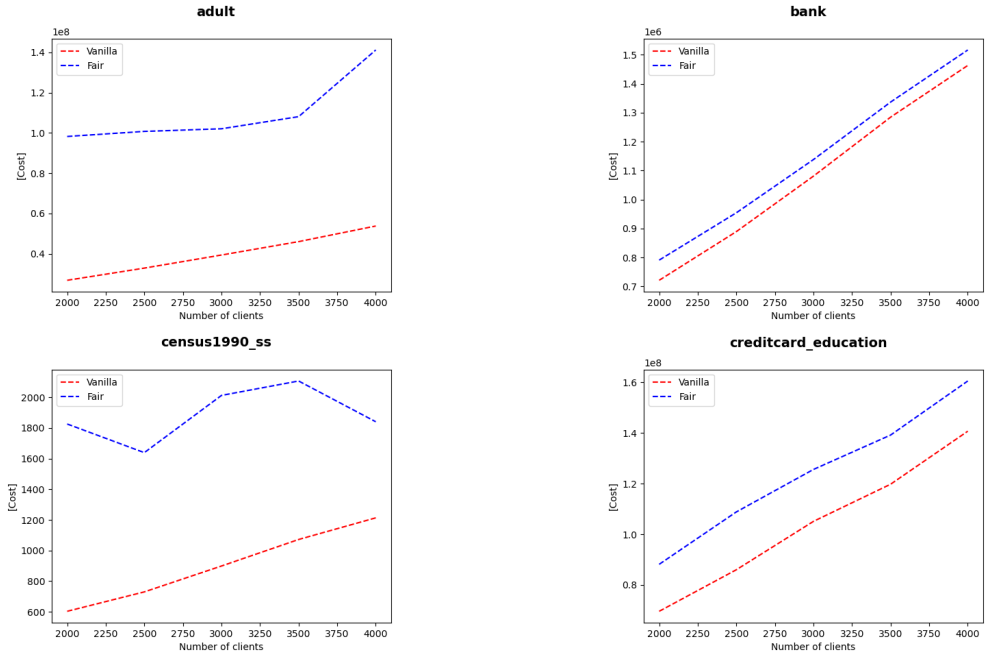

Figure 3: Average cost comparison across multiple datasets for $k = 10$. The top dashed line in all figures in the fair cost and the bottom one is the vanilla cost.

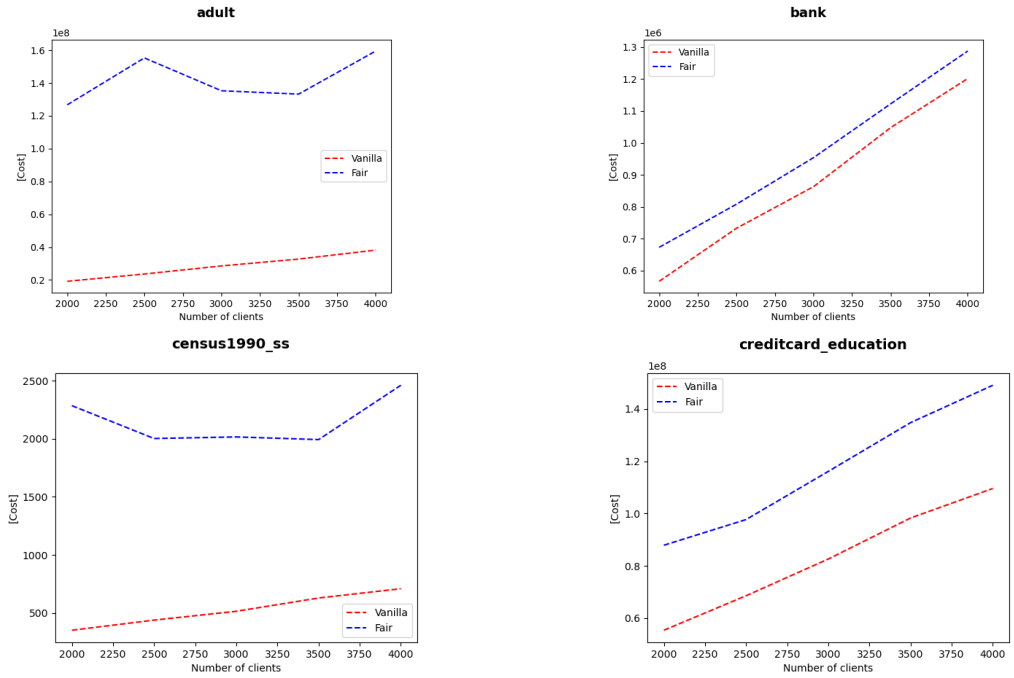

Figure 4: Cost comparison across multiple datasets for $k = 15$. The top dashed line in all figures in the fair cost and the bottom one is the vanilla cost.

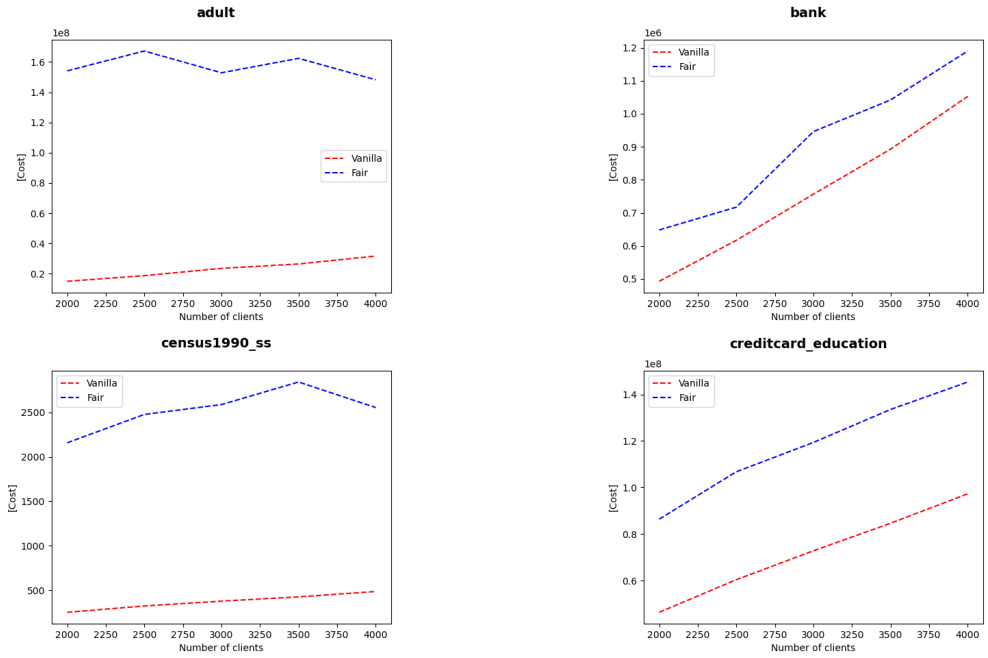

Figure 5: Cost comparison across multiple datasets for $k = 20$. The top dashed line in all figures in the fair cost and the bottom one is the vanilla cost.

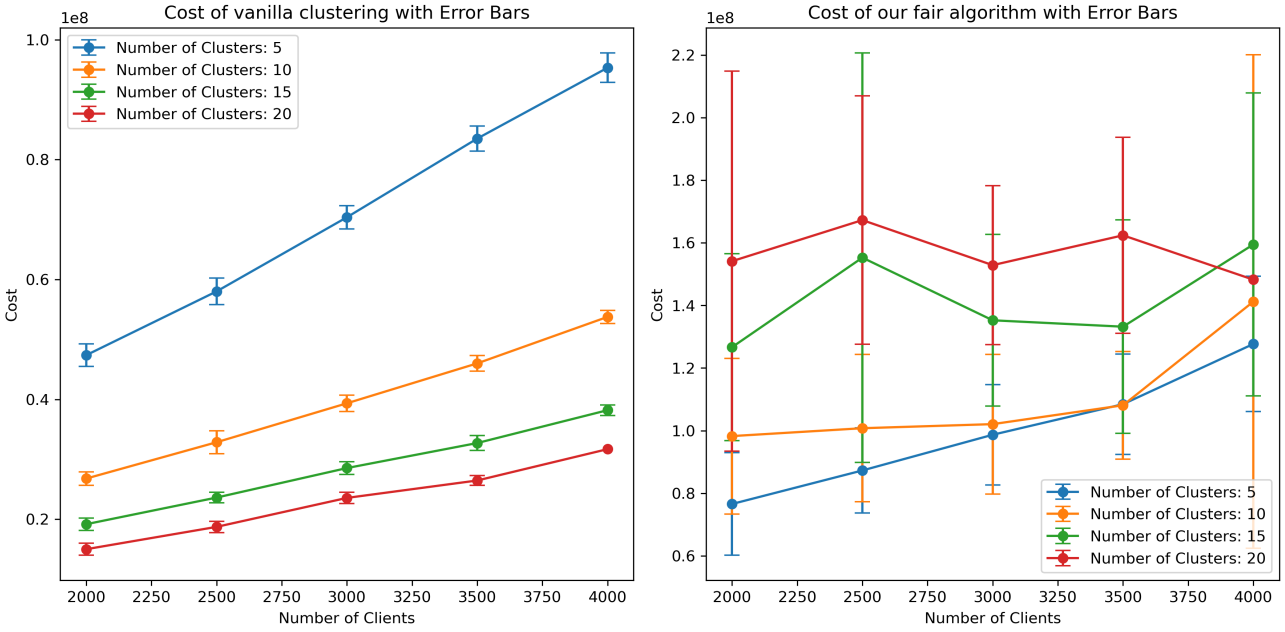

Figure 6: k-MEDIAN cost of the unconstrained and fair algorithms versus k. Each point is the average of 10 trials, with error bars indicating the standard deviation for the dataset *adult_race*. In the left graph, the lines represent cluster count of 5, 10, 15, and 20, top to bottom; while in the right graph, at the left most point, the lines represent 20, 15, 10, and 5 cluster count, top to bottom.

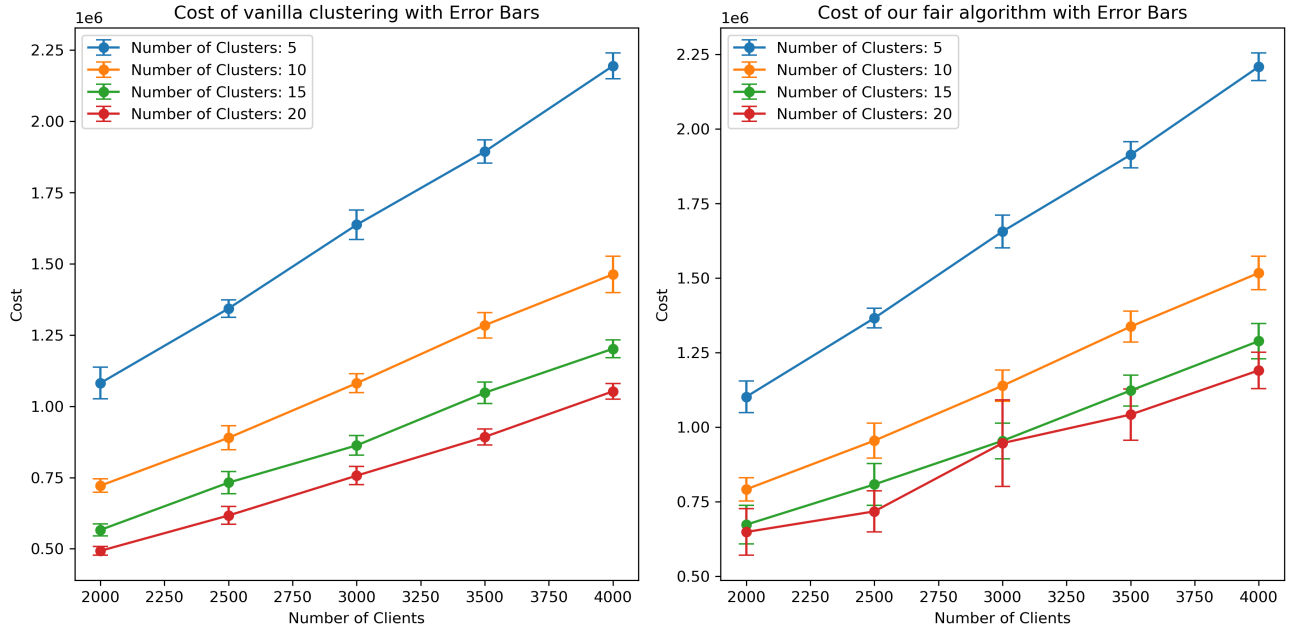

Figure 7: k-MEDIAN cost of the unconstrained and fair algorithms versus k. Each point is the average of 10 trials, with error bars indicating the standard deviation for the dataset *bank*. In the left graph, the lines represent cluster count of 5, 10, 15, and 20, top to bottom; while in the right graph, at the right most point, the lines follow the same pattern.

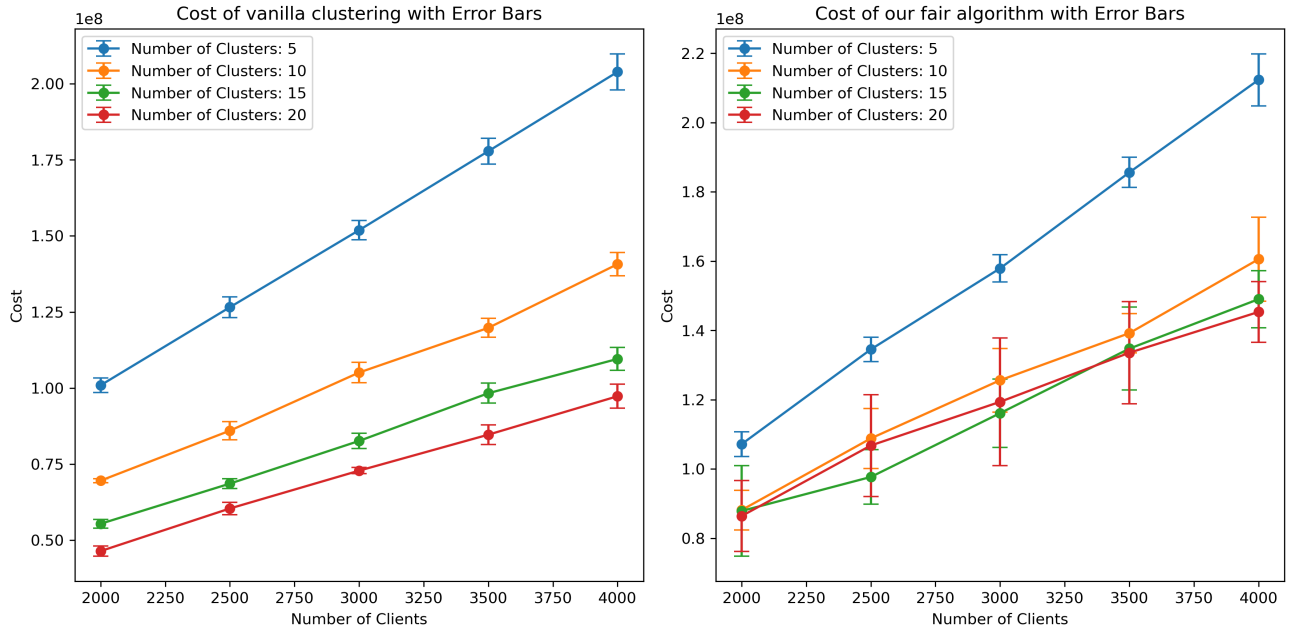

Figure 8: k-MEDIAN cost of the unconstrained and fair algorithms versus k. Each point is the average of 10 trials, with error bars indicating the standard deviation for the dataset *creditcard_education*. In the left graph, the lines represent cluster count of 5, 10, 15, and 20, top to bottom; while in the right graph, at the right most point, the lines follow the same pattern.

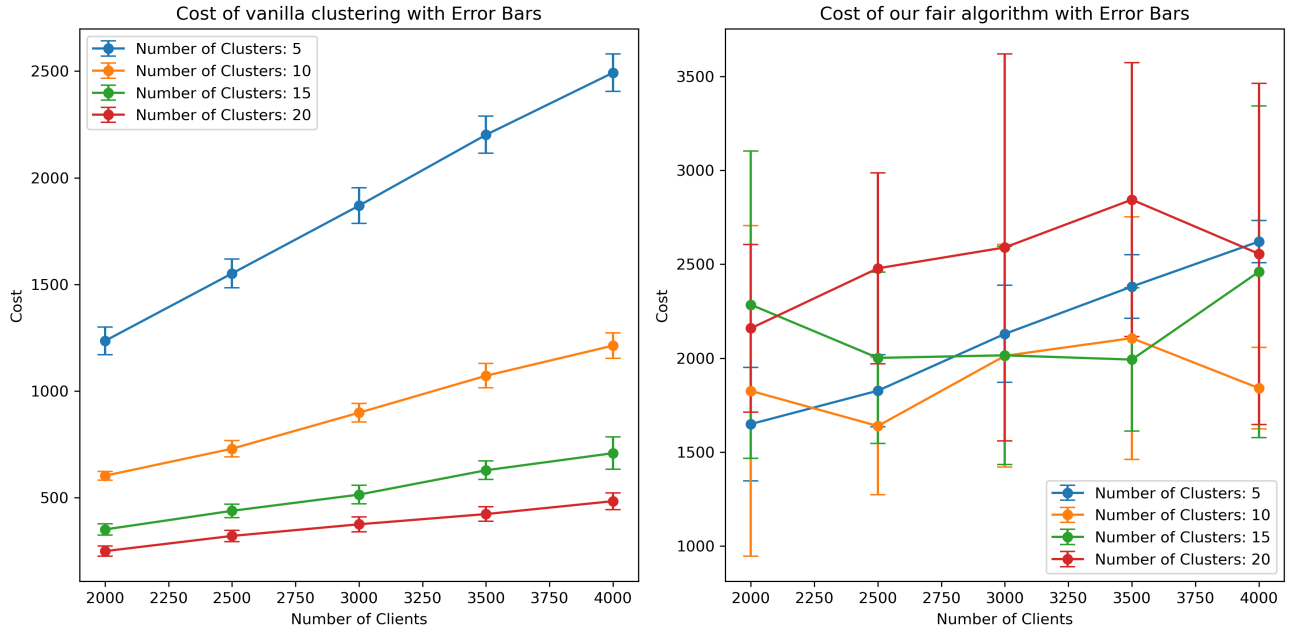

Figure 9: k-MEDIAN cost of the unconstrained and fair algorithms versus k. Each point is the average of 10 trials, with error bars indicating the standard deviation for the dataset *census*1990_*ss*. In the left graph, the lines represent cluster count of 5, 10, 15, and 20, top to bottom; while in the right graph, at the left most point, the lines represent 15, 20, 10, and 5 cluster count, top to bottom.

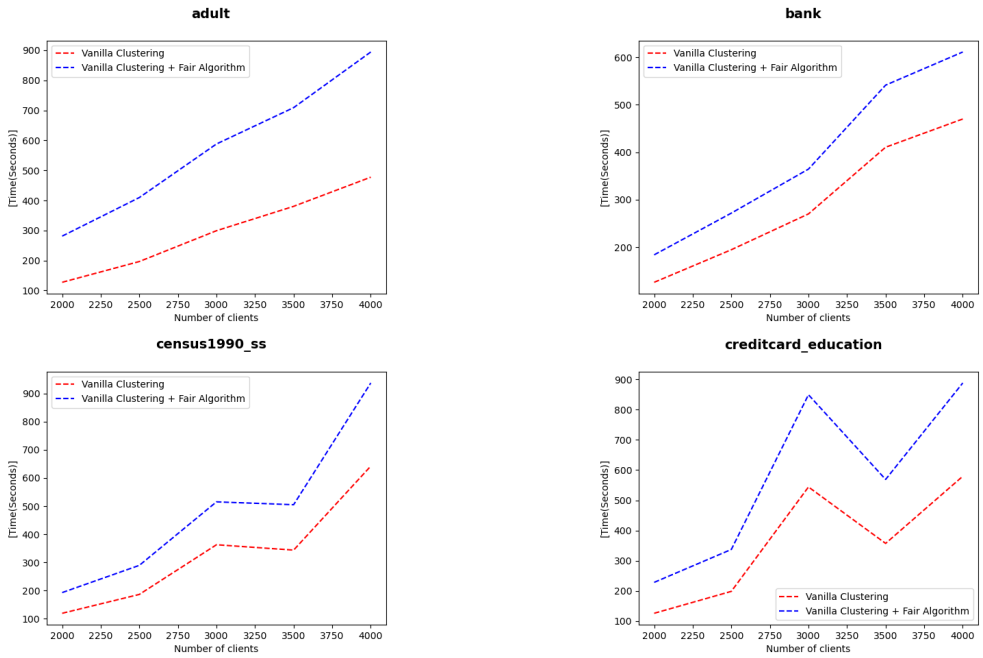

Figure 10: Running time comparison across multiple datasets when $k = 5$. The top dashed line in all figures in the fair cost and the bottom one is the vanilla cost.

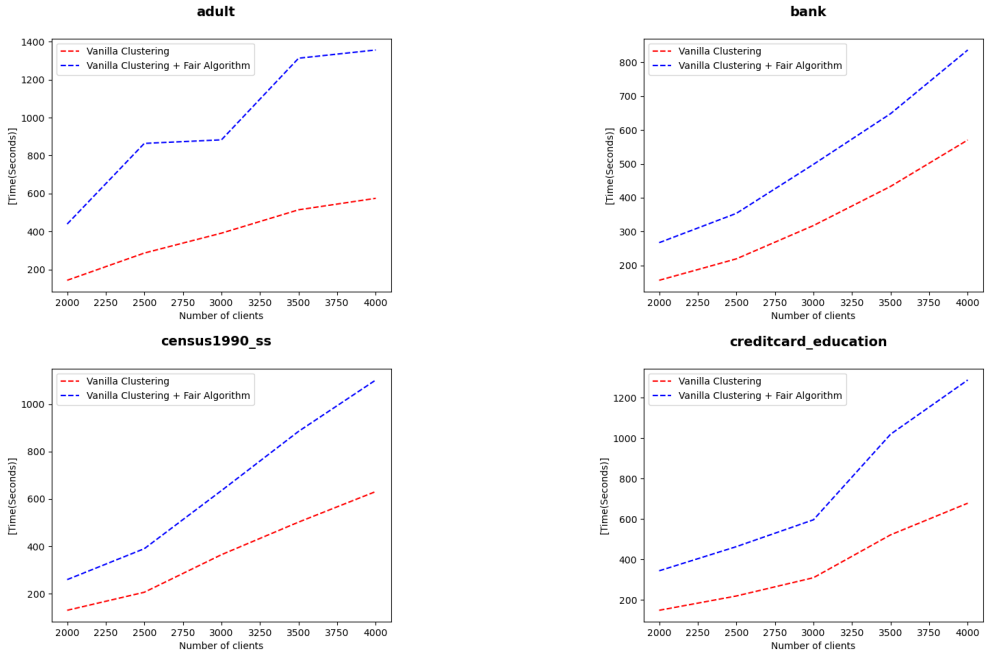

Figure 11: Running time comparison across multiple datasets when $k = 10$. The top dashed line in all figures in the fair cost and the bottom one is the vanilla cost.

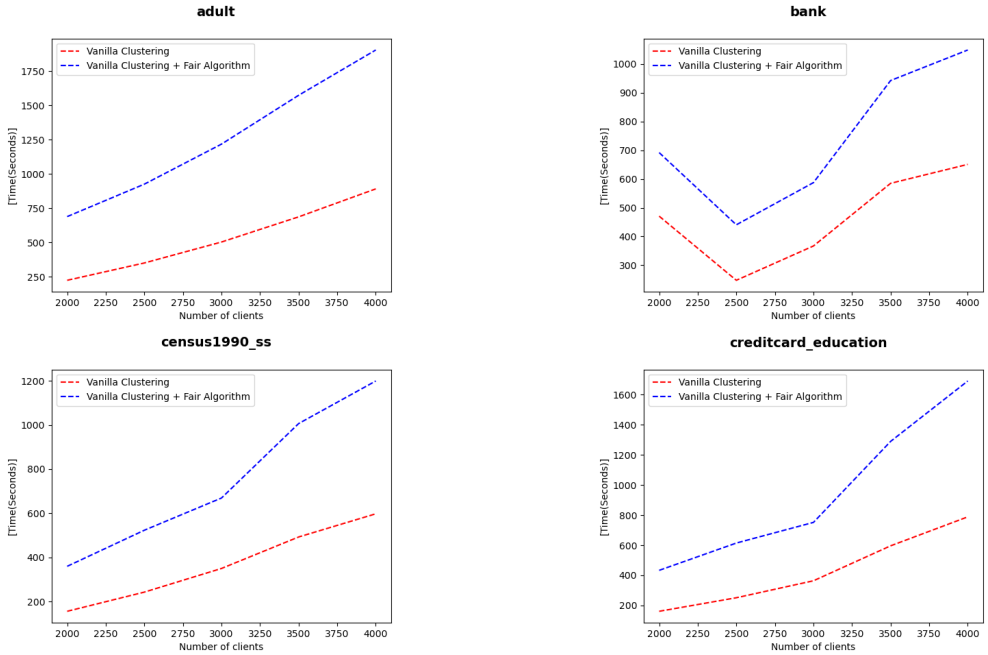

Figure 12: Clustering time comparison across multiple datasets when $k = 15$. The top dashed line in all figures in the fair cost and the bottom one is the vanilla cost.

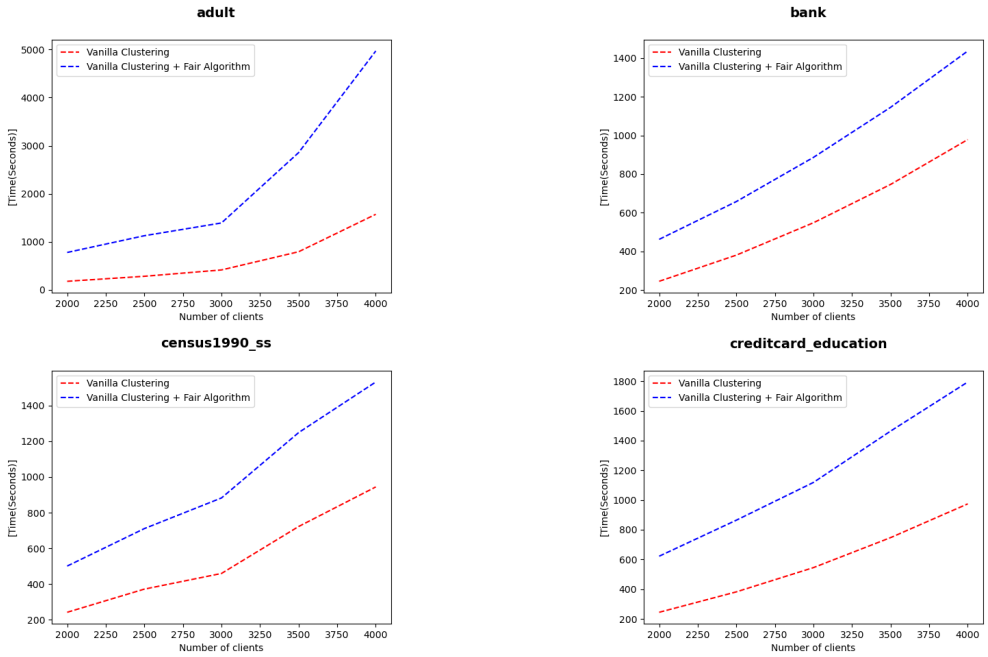

Figure 13: Clustering time comparison across multiple datasets when $k = 20$. The top dashed line in all figures in the fair cost and the bottom one is the vanilla cost.

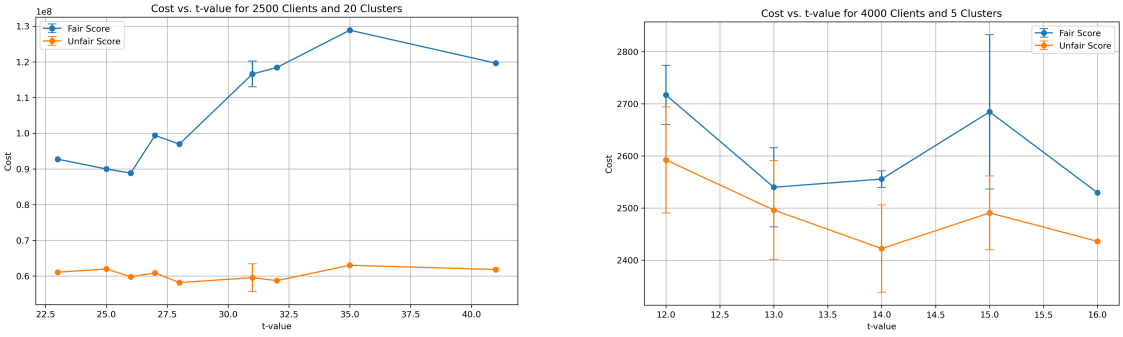

Figure 14: Plot of the fairness-cost trade-off across three distinct experimental runs, left for *creditcard_education* Dataset ($k = 5$ and number of clients $= 1000$) and bottom for *census*1990_*ss* ($k = 5$ and number of clients $= 4000$). The x-axis shows the balance parameter t, and the y-axis represents the clustering cost. The solid blue (top) line tracks the cost of the final fair solution, while the orange (bottom) line represents the baseline cost of the vanilla (unfair) clustering for each run.