Service in Your Neighborhood: Fairness in Center Location

Christopher Jung
University of Pennsylvania, Philadelphia, PA, USA
chrjung@seas.upenn.edu

Sampath Kannan
University of Pennsylvania, Philadelphia, PA, USA
kannan@cis.upenn.edu

Neil Lutz
Iowa State University, Ames, IA, USA
nlutz@istate.edu

Abstract

When selecting locations for a set of centers, standard clustering algorithms may place unfair burden on some individuals and neighborhoods. We formulate a fairness concept that takes local population densities into account. In particular, given k centers to locate and a population of size n, we define the "neighborhood radius" of an individual i as the minimum radius of a ball centered at i that contains at least n/k individuals. Our objective is to ensure that each individual has a center that is *within at most a small constant factor of her neighborhood radius.*

We present several theoretical results: We show that optimizing this factor is NP-hard; we give an approximation algorithm that guarantees a factor of at most 2 in all metric spaces; and we prove matching lower bounds in some metric spaces. We apply a variant of this algorithm to real-world address data, showing that it is quite different from standard clustering algorithms and outperforms them on our objective function and balances the load between centers more evenly.

2012 ACM Subject Classification Theory of computation → Theory and algorithms for application domains

Keywords and phrases Fairness, Clustering, Facility Location

Digital Object Identifier 10.4230/LIPIcs.FORC.2020.5

Related Version A full version of the paper is available at `https://arxiv.org/abs/1908.09041`.

Funding *Christopher Jung*: Research supported by a grant from the Quattrone Center, University of Pennsylvania.
Sampath Kannan: Research supported in part by NSF Grants CCF-1733794 and CCF-1763307.

Acknowledgements We thank Moni Naor and Omer Reingold for helpful discussions, and we thank Anupam Gupta for alerting us to the similarities between our Theorem 2 and Lemma 1 of Chan, Dinitz, and Gupta [7].

1 Introduction

Fairness in decision making has become an important research topic as more and more classification decisions, such as college admissions, bank loans, parole and sentencing, are made with the assistance of machine learning algorithms [10]. Such decisions are made on individuals at a particular point in time, and although they have long-term consequences on the individuals affected, there is at least the prospect of these decisions being revisited with new data about these individuals. In contrast, certain infrastructural decisions, for example, about where to locate hospitals, schools, library branches, police stations, or fire stations have long-term consequences on all the residents of a town, district, or county. Stories about

1st Symposium on Foundations of Responsible Computing (FORC 2020).
Editor: Aaron Roth; Article No. 5; pp. 5:1–5:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

neighborhoods not being adequately served make frequent headlines. Such stories range from food deserts in inner cities because of the absence of supermarkets that sell fresh fruit and vegetables to lack of access to medical services in rural areas [14, 26].

In many situations, equal treatment of all individuals requires clustering individuals into roughly equal-sized groups and allocating the same amounts of resources (such as schools or hospitals) to each group. This means that these resources will naturally be located farther away on average from residents of a sparse district. This is generally accepted by society, and one could argue that it is in fact the just way to allocate resources. Thus, even where all individuals are entitled to equal treatment, it is admissible, even desirable, to discriminate based on geographic location. However, even in situations where there is great geographic variation in density, and concomitant variation in how resources are allocated, we would like to ensure some form of fairness to each individual.

We ask what might be a fair way to locate k hospitals, say, in an area with varying population densities. A standard formulation such as the k-center problem is problematic for at least two reasons:

First, in a good k-center solution, a hospital located in an urban area would be overcrowded. Thus, one kind of fairness we want is *load balance*; the numbers of people served by each center should be as close to equal as possible. Intuitively the definition we give seems tailored to provide such balance, and we confirm this empirically.

Second, people living in areas with different population densities have different expectations for a reasonable distance to travel to a hospital. In rural areas, it would be unreasonable for a resident to expect to find a hospital within a mile, say, of her residence, but in an urban area this might be an entirely reasonable expectation. This is reinforced by the fact that individuals in dense urban areas – especially dense, low-income urban neighborhoods – are more likely to rely on bicycles or public transit and less likely to have access to a car [24].

Taking this perspective, consider the problem of serving a population P of n people using k centers, for a given k. On average, we expect each center to serve n/k people. An individual i might reasonably hope that the center that serves i is no farther than the $(\lceil \frac{n}{k} \rceil)^{\text{th}}$ nearest individual from i, including i itself. Thus, for a given P and k, we define the *neighborhood radius* $NR(i)$ to be the distance from i to its $(\lceil \frac{n}{k} \rceil - 1)^{\text{th}}$ nearest neighbor.

Unfortunately, it is not always possible to find a solution with k centers where each individual finds a center within her neighborhood radius. Hence our goal is to optimize how far we deviate from this ideal. Given a solution S that specifies the placement of the k centers, let $d(i, S)$ denote the distance from individual i to the closest center in S. Let

$$\alpha(S) = \max_i \frac{d(i,S)}{NR(i)}$$

denote the maximum factor by which an individual's distance to the center nearest to her, exceeds her neighborhood radius. We say that an algorithm achieves α-fairness if the solution S it produces has $\alpha(S) \leq \alpha$. The goal of this paper is to design an efficient algorithm to locate k centers that achieves a small value of α, the maximum factor by which any individual's fair expectations are not met.

Fair k-Center: For as small a value of α as feasible, given n points in a metric space, and a number k, find a solution S^* consisting of a subset of at most k of the given points so that $\alpha(S^*) \leq \alpha$.

One could formulate a Steiner version of this problem, where centers are allowed at arbitrary points in the metric space, but we do not consider this variant in this paper. We also formulate an extremal version of the problem: For a given metric space, what is the worst-case value, over all possible configurations of points in the metric space, of $\alpha(S^*)$? We perform empirical comparisons between our fair k-center formulation and the standard k-center, k-means, and k-medians formulations. Using algorithms designed for each of these optimization problems, we select sets of center locations based on two geographical data sets from Fairfax County, Virginia and Allegheny County, Pennsylvania.

Our results are as follows:

- There is an efficient algorithm that achieves $\alpha = 2$ for any set of points and any parameter k, in any metric space (Theorem 2). We have come to learn that the same algorithm was discovered earlier in a different context by [7].
- Finding the optimal α for a given set of points and parameter k is NP-hard (Theorem 8).
- There are metric spaces and configurations of points for which $\alpha = 2$ is the best possible (Proposition 6). For Euclidean spaces there are configurations that require $\alpha = \sqrt{2}$ (Proposition 7).
- On real data standard clustering algorithms achieve worse α than is achieved by an algorithm we describe (Table 1).
- Associating with any algorithm a vector of at most k values giving the number of points assigned to each center, and viewing load balance as the variance of this vector, our algorithm empirically achieves much better load balance than the other clustering algorithms (Table 2).

1.1 Related Work

There is a rapidly growing body of literature on fair clustering [9, 17, 18, 3, 8, 6, 5]. Most of this work has attempted to optimize standard k-center, k-means, and k-medians objective functions, but under some fairness constraints. In particular, there has been a focus on group fairness: requiring that each group must have approximately equal representation across clusters. Two of the motivations for our fairness notion are that outliers should not disproportionately affect clustering outcomes, and that cluster sizes should be roughly equal. These motivations are shared, respectively, by density-based clustering [27, 1], in which data points in sparse regions are treated as noise, and by load-balanced clustering [19].

2 Defining α-fairness

We consider a nonempty collection P of (not necessarily distinct) points in a metric space (X, d) and some positive integer parameter $k \leq |P| = n$. A *centers algorithm* takes an instance (P, k) as input and returns a set $S \subseteq P$ with $|S| \leq k$ of designated *centers*. The travel distance from a point $x \in X$ to S is $d(x, S)$, the minimum distance from x to a center in S:

$$d(x, S) = \min\{d(x, s) : s \in S\}\,.$$

The goal of a centers algorithm is to select a "good" set of centers according to some criterion. For example, the following are well-studied optimization problems based on natural objective functions for assessing the quality of a solution set.

- ***k*-Center:** minimize the maximum travel distance among individuals in P, $\max_{i \in P} d(i, S)$ [2].
- ***k*-Medians:** minimize the average travel distance, or equivalently, $\sum_{i \in P} d(i, S)$ [16].
- ***k*-Means:** minimize the sum of the squares of the travel distances, $\sum_{i \in P} d(i, S)^2$ [21].

The objective function we introduce, unlike those above, is based on the *neighborhood radius* at a point $x \in X$, $NR(x)$. That is, the minimum radius r such that at least $|P|/k$ of the points in P are within distance r of x:

$$NR_{P,k}(x) = \min\{r : |B_r(x) \cap P| \geq n/k\} ,$$

where $B_r(x)$ is the closed ball of radius r around x. When P and k are clear from context, we omit these subscripts and simply denote the neighborhood radius at x by $NR(x)$.

We quantify the fairness of a set of centers S on a set of points according to the worst ratio between travel distance and neighborhood radius for any point in P:

$$\alpha_{P,k}(S) = \sup_{i \in P} \frac{d(i,S)}{NR_{P,k}(i)} ,$$

adopting the conventions that $0/0 = \infty/\infty = 1$ and $c/0 = \infty$ for any $c > 0$.

Given a centers algorithm A and a constant α, we say that A achieves α*-fairness* on an instance (P, k) if

$$\alpha_{P,k}(A(P,k)) \leq \alpha .$$

We say that A is α*-fair* in the given metric space (X, d) if it achieves α-fairness on every instance. That is, if $\alpha_{P,k}(A(P,k)) \leq \alpha$ for all $P \subseteq X$ and all $1 \leq k \leq n$.

Solutions that are optimal for other standard objective functions can be infinitely unfair with respect to α, as shown by the following example on the real line.

▶ **Example 1.** Let $k = 3$ and consider $P = \{-x, 0, 0, 1, 1, x\}$, where x is some large number, as pictured in Figure 1. The optimal solution with respect to the k-center, k-medians, and k-means objective functions is to place one center at either 0 or 1 and the other two centers at $-x$ and x. But the neighborhood radius is 0 at 0 and 1, and whichever of these is not chosen as a center will have to travel a distance of 1 to the nearest center, meaning that

$$\alpha(\{-x, 0, x\}) = \alpha(\{-x, 1, x\}) \geq \frac{1}{0} = \infty .$$

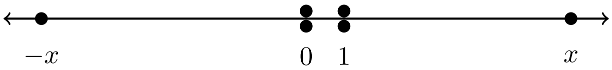


Figure 1 For the above population with $k = 3$, optimizing for $\max_{i \in P} d(i, S)$, $\sum_{i \in P} d(i, S)$, or $\sum_{i \in P} d(i, S)$ will yield a solution that is not α-fair for any finite α.

Although we do not consider allowing Steiner points as centers in this paper, it is clear that even optimal Steiner solutions to the three classical problems – all of which place only one center in the interval $[0, 1]$ – do no better in terms of our fairness objective. This example demonstrates that a different approach is needed to achieve even the weakest of fairness guarantees. In the appendix, we include two more examples of metric spaces in which strong fairness guarantees are easy to achieve.

3 Theoretical Results

Given an instance (P, k), let $\alpha^*_{P,k}$ be the minimum value such that α-fairness can be achieved. In this section we prove that

$$1/2 \leq \alpha^*_{P,k} \leq 2$$

always holds, that equality is possible at each end of that bounding interval, and that $\alpha^*_{P,k}$ is NP-hard to compute.

3.1 A 2-Fair Algorithm

We now give an algorithm, 2FairKCenter, that achieves 2-fairness on every instance and in every metric space. In each iteration, 2FairKCenter chooses a center s with minimum neighborhood radius among the set Z of candidate centers. Then, it removes from Z all points i that are sufficiently close to s.

We have recently become aware that achieving 2-fairness is equivalent to finding a (k/n)-density net, as defined by Chan, Dinitz, and Gupta in the context of constructing slack spanners [7]. In proving that ϵ-density nets can be found in polynomial time for all $\epsilon \in (0,1)$, that work describes an algorithm that is essentially identical to 2FairKCenter. In order to keep this paper self-contained, we include the algorithm description and proof of 2-fairness here.

Algorithm 1 2FairKCenter(P, k).

```
$Z = P$
$S = \emptyset$
while $S \neq \emptyset$ do
    choose $s \in \arg\min_{i \in Z} NR_{P,k}(i)$
    $S = S \cup \{s\}$
    $Z = \{i \in Z : d(i,s) > NR_{P,k}(i) + NR_{P,k}(s)\}$
end
return $S$
```

▶ **Theorem 2.** *2FairKCenter is 2-fair in every metric space.*

Proof. Fix a metric space (X, d), let $P \subseteq X$, and let $k \leq n$. Let s_j be the j^{th} center added to S. For any $j' > j$, the definition of the set S guarantees that $B_{NR(s_j)}(s_j)$ and $B_{NR(s_{j'})}(s_{j'})$ are disjoint. Thus, the balls

$$B_{NR(s_1)}(s_1), B_{NR(s_2)}(s_2), \dots$$

are all pairwise disjoint, and by the definition of neighborhood radius, each includes at least n/k points in P. It follows that there can be at most k centers, so this algorithm will output a valid solution.

Now, a point $i \in P$ is excluded from Z only when there is some $s \in S$ such that $d(i,s) \leq NR(i) + NR(s)$, which is at most $2 \cdot NR(i)$ by our choice of s. So when our algorithm terminates, $d(i,S)/NR(i) \leq 2$ holds for all $i \in P$. Thus, 2-fairness is achieved on (P,k), and the algorithm is 2-fair. ◀

3.2 Lower Bounds

We now give four lower bounds on fairness: We prove that it is never possible to achieve better than 1/2-fairness on any instance; that no algorithm can be better than 1-fair, regardless of the metric space; that there exist metric spaces in which no algorithm can be better than 2-fair; and that no algorithm can be better than $\sqrt{2}$-fair in Euclidean spaces of dimension greater than 1. The first three of these results demonstrate the tightness of Example 9, Example 10 (See appendix for these examples.), and Theorem 2, respectively. We defer the proofs for the following propositions to the appendix

▶ **Proposition 3.** *In every metric space (X, d), for all $S \subseteq P \subseteq X$ and $1 \leq k \leq |S|$, we have $\alpha_{P,k}(S) \geq 1/2$.*

Combining Theorem 2 with Proposition 3 immediately yields the following.

▶ **Corollary 4.** *2FairKCenter is a 4-approximation algorithm to the best α achievable for any configuration of points in any metric space.*

▶ **Proposition 5.** *For all metric spaces (X, d) and all $\alpha < 1$, there is no centers algorithm that is α-fair in (X, d).*

▶ **Proposition 6.** *There exists a metric space (X, d) such that for all $\alpha < 2$, there is no centers algorithm that is α-fair in (X, d).*

▶ **Proposition 7.** *For all $m \geq 2$ and all $\alpha < \sqrt{2}$, there is no centers algorithm that α-fair in m-dimensional Euclidean space.*

3.3 NP-Completeness

The k-center, k-medians, and k-means problems are all known to be NP-hard, and the problem of checking, for a given instance, whether a given value of the objective function is achievable, is NP-complete [11, 22, 15]. We now show in the following theorem that the same is true for our objective function α.

▶ **Theorem 8.** *The problem of determining whether 1-fairness can be achieved on a given instance is NP-complete.*

Proof. This problem is a special case of the hitting set problem, where the sets are

$$S_i = \{j \in P : d(i, j) \leq \alpha \cdot NR_{P,k}(j)\}$$

for each $i \in P$. So it belongs to NP.

We prove NP-hardness by reduction from the dominating set problem. Let G be a graph on a set U of n vertices, and let $1 \leq k \leq n$; without loss of generality, we assume that $n - k$ is even. We construct a new graph G' that contains G as a subgraph and also has the following:

- a set V of $2n$ vertices with degree 1 such that each vertex $u \in U$ is adjacent to two vertices $u_1, u_2 \in V$, and
- a set W of $6n - 6k$ vertices arranged as $\frac{3}{2}(n - k)$ disjoint 4-cycles.

Letting $P = U \cup V \cup W$ and $k' = 3n - 2k$, we will show that G has a dominating set of size k if and only if there is a set $S \subseteq P$ with $|S| = k'$ and $\alpha_{P,k'}(S) \leq 1$. Since (G', k') can be efficiently computed from (G, k), this will suffice to prove the theorem.

Suppose that G has a dominating set D of size k, and consider the set of centers $S = D \cup T$, where T is a set consisting of two vertices from each of the squares in W, so that $d(w, S) = d(w, T) \leq 1$ for all $w \in W$. Now,

$$\frac{n}{k'} = \frac{n + 2n + 6n - 6k}{3n - 2k} = 3\,,$$

so $NR_{P,k'}(w) = 1$ for each $w \in W$. Each $u \in U$ has at least two neighbors – namely, u_1 and u_2 – so we also have $NR_{P,k'}(u) = 1$ for all $u \in U$, and it follows immediately that $NR_{P,k'}(v) = 2$ for each $v \in V$. The fact that $D \subseteq S$ is a dominating set means that $d(u, S) \leq 1$ for each $u \in U$ and therefore that $d(v, S) \leq 2$ for each $v \in V$. Thus, $\alpha_{P,k'}(S) = 1$.

Conversely, suppose that there is some set $S \subseteq P$ with $|S| = k'$ and $\alpha_{P,k'}(S) \le 1$. Then S must contain at least two vertices from each square in W, so letting $Y = S \cap (U \cup V)$, we have

$$|Y| \le k' - (3n - 3k) = k\,.$$

For each $u \in U$, we must have

$$d(u, Y) = d(u, S) \le NR_{P,k'}(u) = 1\,.$$

We construct a set D by taking each vertex in $Y \cap V$ and replacing it with the adjacent vertex in U. Then $|D| \le |Y| \le k$, and for each $u \in U$, $d(u, D) \le d(u, Y)$, meaning that D is a dominating set for G. We conclude that this problem is NP-complete. ◀

4 Experiments

In this section we measure the fairness of three standard clustering algorithms on two geographic data sets, and we compare their performance to that of a modified version of 2FairKCenter that still guarantees 2-fairness but attempts to be even fairer.

4.1 A Heuristic Refinement of 2FairKCenter

Our algorithm 2FairKCenter always yields a 2-fair solution, but this solution might be less than optimal and use fewer than k centers. To avoid this situation, we introduce AlphaFairKCenter, a version of 2FairKCenter that is parameterized by a fairness guarantee parameter, α. This algorithm achieves α-fairness on every instance by essentially the same argument we used to prove that Algorithm 1 is 2-fair. The catch is that the output will not necessarily be a valid solution: For $\alpha < 2$, Algorithm 2 may select more than k centers on a given instance (P, k).

Algorithm 2 AlphaFairKCenter(α, P, k).

```
$Z = P$
$S = \emptyset$
while $S \neq \emptyset$ do
    choose $s \in \arg\min_{i \in Z} NR_{P,k}(i)$
    $S = S \cup \{s\}$
    $Z = \{i \in Z : d(i, s) > \alpha \cdot NR_{P,k}(i)\}$
return $S$
```

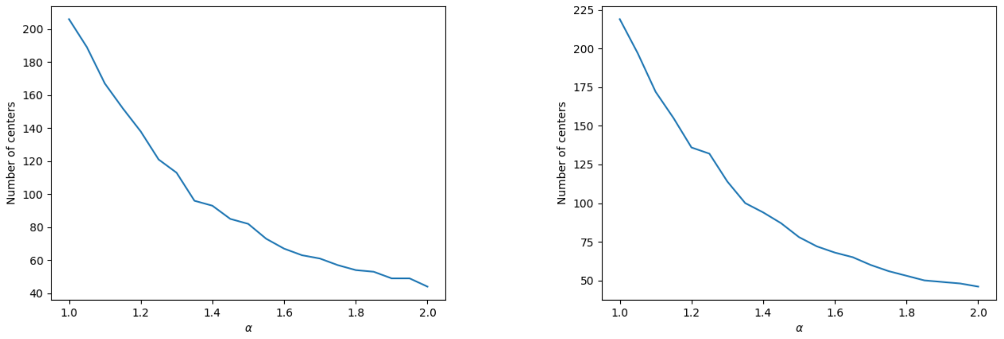
For each instance (P, k), we define a function $f_{P,k} : [1, 2] \to \mathbb{N}$ by

$$f_{P,k}(\alpha) = |\textsc{AlphaFairKCenter}(\alpha, P, k)|\ ,$$

the number of centers chosen by AlphaFairKCenter with parameter α on instance (P, k). Our goal is to find a small α such that $f_{P,k}(\alpha) \le k$. In order to do this, we will perform a binary search on the interval $[1, 2]$, recursively searching the lower half of the interval when $f_{P,k}(\alpha) > k$ and the upper half of the interval otherwise.

If $f_{P,k}$ is a monotonic function, then this search will find

$$\inf\{\alpha \in [1, 2] : f_{P,k}(\alpha) \le k\}$$



(a) Fairfax County, Virginia.

(b) Allegheny County, Pennsylvania.

Figure 2 The number of centers chosen by AlphaFairKCenter for different values of α with $k = 100$ given address points in two counties.

up to arbitrary precision. Intuitively, $f_{P,k}$ has a general tendency to be decreasing – a weaker fairness guarantee requires fewer centers – but in fact $f_{P,k}$ is not necessarily monotonic, and local extrema may cause our search to select a larger α than is necessary.

Fortunately, as shown in Figure 2, $f_{P,k}$ seems to behave monotonically at coarse scales on real data. Furthermore, deviations from monotonicity cannot affect the validity of the solution we find, only its optimality. Hence, this binary search appears to be a useful heuristic, and we employ it in our algorithm FairKCenter. In addition to an instance (P, k), this algorithm takes as input a precision parameter t that determines the depth of the binary search.

Algorithm 3 FairKCenter(t, P, k).

```
$low = 1$
$high = 2$
for $i = 1, 2, \ldots, t$ do
    $mid = (low + high)/2$
    if $|\text{AlphaFairKCenter}(mid, P, k)| \leq k$ then
        $high = mid$
    else
        $low = mid$
return AlphaFairKCenter$(high, P, k)$
```

4.2 Experimental Setup

We applied our algorithm FairKCenter to select 100 center locations in two American counties: Fairfax County, Virginia, and Allegheny County, Pennsylvania. Fairfax County is located near Washington, D.C., and is primarily suburban. According to the 2010 United States Census [25], its population density is 1068 people per square kilometer, with census tracts ranging in density from 56 to 23,397 people per square kilometer. Allegheny County contains the city of Pittsburgh as well as many of its suburbs and exurbs; in the 2010 Census, the county's population density was 647 people per square kilometer, with census tracts ranging in density from 48 to 12,474 people per square kilometer.[1]

[1] The stated ranges of population density exclude the few census tracts with fewer than 100 people. Some census tracts are uninhabited.

The “populations” for our experiment were the sets of all address points in each county, not the locations of individual people. The Fairfax data set contains 537,514 address points, and the Allegheny data set contains 370,776. The data sets were published by Fairfax County GIS and the Allegheny County / City of Pittsburgh / Western PA Regional Data Center, respectively [13, 23]. We measured Euclidean distance after projecting (latitude, longitude) pairs onto the plane using the Universal Transverse Mercator coordinate system.

The bottleneck for our algorithm in terms of running time is calculating the neighborhood radius for each point. In order to accelerate this process, we used the Python library KD-tree [4]. The KD-tree data structure allows us to quickly query the distance to any point's $(\lceil n/k \rceil - 1)^{\text{th}}$ nearest neighbor, which is exactly the definition of the neighborhood radius at that point.

We compared the performance of FairKCenter to standard algorithms for the k-means, k-medians, and k-center problems:

- For k-means, we used the Python library sklearn, which employs either Lloyd's algorithm [20] or Elkan's algorithm [12], depending on the problem size and parameters.
- For k-medians, we used the Python library pyclustering to execute a variant of Lloyd's algorithm that calculates a median instead of a centroid in each iteration.
- For k-center, we implemented the standard greedy approximation algorithm [28].

For each county, we assessed the performance of each algorithm according to our fairness objective function α as well as the k-means, k-medians, and k-center objective functions. We also measured how well each algorithm balanced the load by finding the standard deviation in the number of addresses served by each center.

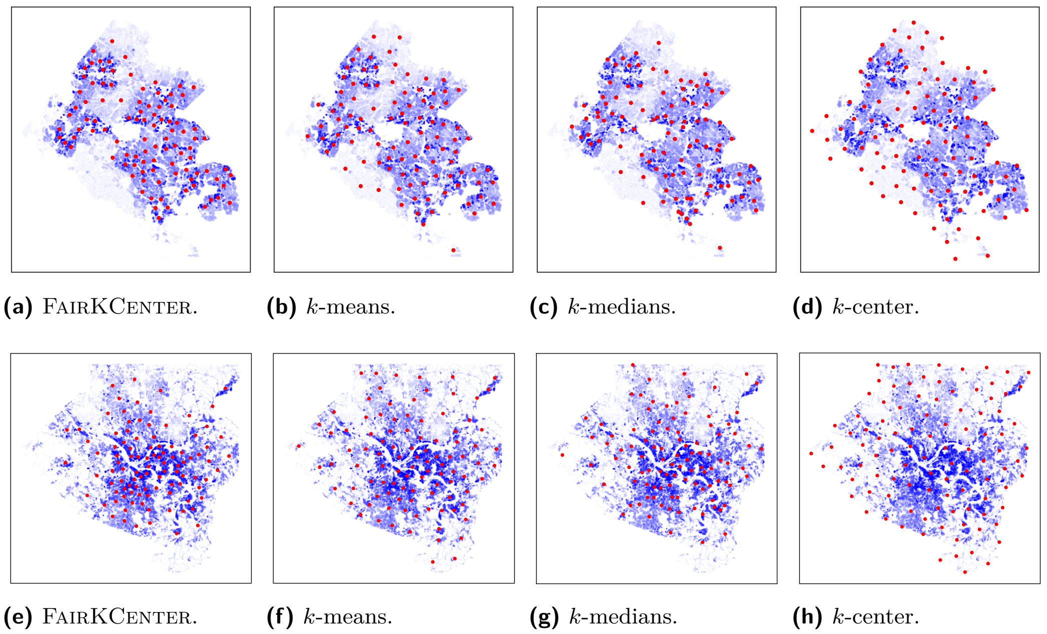
4.3 Experimental Results

In Figure 3, we show the population density map of Fairfax County and Alllegheny County along with 100 centers whose locations were determined by FairKCenter, k-means, k-medians, and k-center. Each tiny blue point, whose transparency has been slightly lowered in order to better show the population density of each region, corresponds to an address point, and each red point is a center.

In Table 1, we show how each algorithm performs in terms of each problem's objective function for each dataset. The values are in units of meters for the k-medians and k-center objective functions, and square meters for the k-means objective function.

Table 1 Performance of each algorithm on Fairfax County and Allegheny County with respect to various objective functions.

		Objective function			
	Algorithm	α	k-means	k-medians	k-center
Fairfax	FairKCenter	1.34306	1811007	1373.79	10002.64
	k-means	1.45643	1137163	1217.07	5662.06
	k-medians	1.80263	1613910	1393.39	6446.81
	k-center	2.57986	2027176	1675.85	2925.80
Allegheny	FairKCenter	1.33721	3600461	1902.78	11615.59
	k-means	1.57453	2082104	1632.00	6183.67
	k-medians	1.90726	3020040	1841.34	7835.79
	k-center	2.67804	3763269	2272.11	3815.03


(a) FairKCenter. (b) k-means. (c) k-medians. (d) k-center.

(e) FairKCenter. (f) k-means. (g) k-medians. (h) k-center.

Figure 3 Placing 100 centers in Fairfax County (a–d) and Allegheny County (e–h), using FairKCenter and algorithms for the k-means, k-medians, and k-center problems.

It is immediately apparent that FairKCenter tends to place more centers in denser regions, compared to other algorithms. This is consistent with the intuition behind the algorithm, as address points in dense regions have relatively smaller neighborhood radius. Although the maximum travel distance is increased significantly relative to the other algorithms, these large travel distances are experienced only by few residents of particularly sparse areas. The increase in average travel distance is more modest, and in Fairfax County our algorithm does even better than the k-medians algorithm with respect to the k-medians objective function. In exchange for these compromises, our algorithm does significantly better with respect to α, ensuring that no individual will needs to venture too far from their density-dependent neighborhood.

Furthermore, FairKCenter balances the load more evenly across centers than other algorithms. Table 2 shows the standard deviation in the number of address points served by each center, i.e., the number of points for which that center is the nearest. For both counties, this value is significantly lower for FairKCenter than for the other algorithms. Our algorithm balances load particularly well compared to the k-center algorithm, which essentially ignores population density.

Table 2 Standard deviation in cluster sizes.

	County	
Algorithm	Fairfax	Allegheny
FairKCenter	1032.49	1696.95
k-means	1344.17	2273.53
k-medians	1630.06	1922.53
k-center	2758.44	5691.10

5 Conclusion

We have formulated a simple geometric concept that captures an intuitive notion of fairness: To whatever extent possible, an individual should have access to resources within her own neighborhood. We have proved basic properties of this fairness concept, given a general approximation algorithm for its optimization, and shown that this algorithm performs well on real data.

One potential future direction for this work is to refine the notion of what constitutes an individual's "neighborhood" for a given purpose. We used the inverse of local population density as a proxy for the size of a neighborhood, and there are good reasons to believe that these two are correlated. But a more sophisticated approach to defining neighborhood size – and possibly shape – might incorporate data on transit times and availability of different modes of transportation. More ambitiously, cellular location data might be used to establish the extent of the common "orbits" of residents of a given small area.

On the theoretical side, an obvious next direction is to find algorithms that yield stronger approximation ratios. Our algorithm FairKCenter improved on 2FairKCenter in our experiments by less aggressively eliminating candidate centers, but we do not have any theoretical characterization of the instances on which FairKCenter will achieve fairness that is strictly better 2FairKCenter. The monotonicity property that is necessary to ensure a fully successful binary search does not hold in general, but one might still be able to identify critical values of α in this range, solve the problem at each of these critical values, and use the smallest one that results in a number of centers that is at most k.

Another interesting extension would be to allow Steiner points, removing the restriction that the solution set S is a subset of the population P. While *prima facie* it looks like we might have to consider infinitely many such possible centers, one can show that the only points we need to consider as centers are points x such that for some r, the boundary of $B_r(x)$ contains at least 3 of the input points. This reduces the number of Steiner points to consider to at most n^3.

Finally, while we have tight lower and upper bounds on α for arbitrary metric spaces, for Euclidean spaces we have a lower bound of $\sqrt{2}$ and an upper bound of 2. It would be interesting to close the gap, perhaps by improving the upper bound for this case.

References

1 Amineh Amini, Ying Wah Teh, and Hadi Saboohi. On density-based data streams clustering algorithms: A survey. *J. Comput. Sci. Technol.*, 29(1):116–141, 2014. doi:10.1007/s11390-014-1416-y.

2 T. Asano, B. Bhattacharya, M. Keil, and F. Yao. Clustering algorithms based on minimum and maximum spanning trees. In *Proceedings of the Fourth Annual Symposium on Computational Geometry*, SCG '88, pages 252–257, New York, NY, USA, 1988. ACM. doi:10.1145/73393.73419.

3 Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. Scalable fair clustering. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 405–413, 2019. URL: http://proceedings.mlr.press/v97/backurs19a.html.

4 Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, September 1975. doi:10.1145/361002.361007.

5 Suman Kalyan Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. 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*, pages 4955–4966, 2019. URL: http://papers.nips.cc/paper/8741-fair-algorithms-for-clustering.

6 Ioana Oriana 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, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA*, pages 18:1–18:22, 2019. doi:10.4230/LIPIcs.APPROX-RANDOM.2019.18.

7 T. H. Hubert Chan, Michael Dinitz, and Anupam Gupta. Spanners with slack. In Yossi Azar and Thomas Erlebach, editors, *Algorithms – ESA 2006*, pages 196–207, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

8 Xingyu Chen, Brandon Fain, Liang Lyu, and Kamesh Munagala. Proportionally fair clustering. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 1032–1041, 2019. URL: http://proceedings.mlr.press/v97/chen19d.html.

9 Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5029–5037, 2017. URL: http://papers.nips.cc/paper/7088-fair-clustering-through-fairlets.

10 A. Chouldechova and A. Roth. The Frontiers of Fairness in Machine Learning. *arXiv e-prints*, October 2018. arXiv:1810.08810.

11 Petros Drineas, Alan M. Frieze, Ravi Kannan, Santosh Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56(1-3):9–33, 2004. doi:10.1023/B:MACH.0000033113.59016.96.

12 Charles Elkan. Using the triangle inequality to accelerate k-means. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pages 147–153. AAAI Press, 2003. URL: http://dl.acm.org/citation.cfm?id=3041838.3041857.

13 Fairfax County GIS. Address points, 2019. URL: https://catalog.data.gov/dataset/address-points-b4b16.

14 Heather Haddon and Annie Gasparro. Companies and government seek new answers for food deserts. *The Wall Street Journal*, October 2016. URL: https://www.wsj.com/articles/companies-and-government-seek-new-answers-for-food-deserts-1476670262.

15 Dorit S. Hochbaum. When are NP-hard location problems easy? *Annals OR*, 1(3):201–214, 1984. doi:10.1007/BF01874389.

16 Anil K. Jain and Richard C. Dubes. Algorithms for clustering data, 1988.

17 Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. 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*, pages 3448–3457, 2019. URL: http://proceedings.mlr.press/v97/kleindessner19a.html.

18 Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. 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*, pages 3458–3467, 2019. URL: http://proceedings.mlr.press/v97/kleindessner19b.html.

19 Ying Liao, Huan Qi, and Weiqun Li. Load-balanced clustering algorithm with distributed self-organization for wireless sensor networks. *IEEE sensors journal*, 13(5):1498–1506, 2012.

20 S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*, 28(2):129–137, September 2006. doi:10.1109/TIT.1982.1056489.

21 J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press. URL: https://projecteuclid.org/euclid.bsmsp/1200512992.

22 Nimrod Megiddo and Kenneth J. Supowit. On the complexity of some common geometric location problems. *SIAM J. Comput.*, 13(1):182–196, 1984. doi:10.1137/0213014.

23 Allegheny County / City of Pittsburgh / Western PA Regional Data Center. Allegheny county address points, 2018. URL: `https://catalog.data.gov/dataset/allegheny-county-address-points-07dff`.

24 Issi Romem. Getting around, or just getting by? Where people live with fewer cars, 2019. URL: `https://www.trulia.com/research/people-per-vehicle-map/`.

25 U.S. Census Bureau. Population, housing units, area, and density: 2010 – county – census tract, 2010 census summary file 1, 2010 Census. URL: `https://factfinder.census.gov`.

26 Kelly Virella. Doctors and health workers reflect on rural america's limited access to care. *The New York Times*, 2018. URL: `https://www.nytimes.com/2018/07/19/reader-center/rural-health-care.html`.

27 Wei-Tung Wang, Yi-Leh Wu, Cheng-Yuan Tang, and Maw-Kae Hor. Adaptive density-based spatial clustering of applications with noise (dbscan) according to data. In *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 1, pages 445–451. IEEE, 2015.

28 David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.

A Additional Examples from Section 2

▶ **Example 9** (A star graph)**.** Consider the graph metric on a star graph: a graph with vertex set $X = \{z, x_1, \dots, x_m\}$ and edge set $\{\{z, x_1\}, \dots, \{z, x_m\}\}$, as in Figure 4. If $(m+1)/2 < k < m+1$, then $NR_{X,k}(z) = 1$ and

$$NR_{X,k}(x_1) = \cdots = NR_{X,k}(x_m) = 2\,.$$

As long as $z \in S$, we have $d(z, S) = 0$ and

$$d(x_1, S) = \cdots = d(x_m, S) = 1\,,$$

so an algorithm that selects z as a center achieves 1/2-fairness for any such instance.

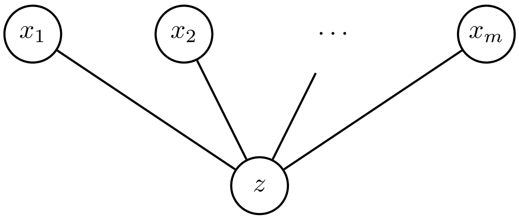


Figure 4 1/2-fairness can be achieved in this instance by selecting z as a center.

▶ **Example 10** (The discrete metric)**.** Consider any nonempty set X under the discrete metric, where $d(x, y) = 0$ if $x = y$ and 1 otherwise. Let $P \subseteq X$ be any nonempty set, let $1 \le k \le n$, let $S \subseteq P$ be any nonempty set of size k, and let $i \in P$. If $k = n$, then we have $NR(i) = d(i, S) = 0$. Otherwise, $NR(i) = 1$ and $d(i, S) \in \{0, 1\}$. Hence, there is a 1-fair algorithm for discrete metric spaces: Select any nonempty set of centers.

Most metric spaces do not share the essential property of Examples 9 and 10: the existence of an extremely central point that is close to all other points. In general, achieving fairness requires more care about how one distributes multiple centers.

An ideal situation for the goal of 1-fairness would if the population were arranged in well-separated "villages" containing n/k individuals each, where the diameter of each village is less than the space between villages. In this case, placing a single center anywhere in each village would achieve 1-fairness.

But consider what happens if two villages are brought closer together: Some of an individual's $\lceil n/k \rceil - 1$ closest neighbors might then reside in the other village, meaning that her neighborhood radius no longer encompasses the entirety of her own village, possibly including that village's center. This general situation is more difficult, but in the one-dimensional case, 1-fairness can still be achieved, as the following example shows.

Algorithm 4 RealLineFairKCenter(P, k).

```
$S = \emptyset$
while $P \neq \emptyset$ do
    $s = \min P$
    $S = S \cup \{s\}$
    $P = P \setminus B_{NR_{P,k}}(s)$
return $S$
```

▶ **Example 11** (The real line). Given any finite set $P \subseteq \mathbb{R}$, Algorithm 4 starts from the left and takes every $\lceil n/k \rceil^{\text{th}}$ point. Notice that the population P in which the neighborhood radius is determined changes with each iteration.

Each iteration removes at least n/k points from P, so the algorithm will terminate with at most k centers. The $\lceil n/k \rceil$ closest points to any point on the line must include the $j\lceil n/k \rceil^{\text{th}}$ smallest point for some $1 \leq j \leq k$, so this algorithm is 1-fair.

Unfortunately, this approach cannot be extended to higher dimensions. As we show in Section 3, 1-fairness is not always achievable, even in the Euclidean plane.

B Proofs of Propositions in Subsection 3.2

▶ **Proposition 3.** *In every metric space (X, d), for all $S \subseteq P \subseteq X$ and $1 \leq k \leq |S|$, we have $\alpha_{P,k}(S) \geq 1/2$.*

Proof. For each center $s \in S$, define the set

$$J(s) = \{i \in P : d(i, s) = d(i, S)\} ,$$

the set of points in P for which s is a closest center. There are at most k centers in S, and $\bigcup_{s \in S} J(s) = P$, so there must be some center $\hat{s}$ with $J(\hat{s}) \geq n/k$. Now, let

$$\hat{\imath} \in \arg\max_{i \in J(\hat{s})} d(i, \hat{s}) .$$

For each $j \in J(\hat{s})$, we know that $d(j, \hat{s}) \leq d(\hat{\imath}, \hat{s})$, so by the triangle inequality, $d(\hat{\imath}, j) \leq 2d(\hat{\imath}, \hat{s})$. Thus, $B_{2d(\hat{\imath},\hat{s})}(\hat{\imath})$ contains all of $J(\hat{s})$, meaning that

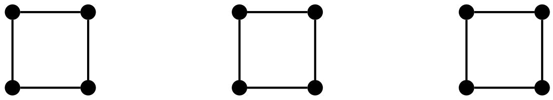
$$\left|B_{2d(\hat{\imath},\hat{s})} \cap P\right| \geq |J(\hat{s})| \geq n/k ,$$

so $NR_{P,k}(\hat{\imath}) \leq 2d(\hat{\imath}, \hat{s}) = 2d(\hat{\imath}, S)$. It follows that $\alpha_{P,k}(S) \geq 1/2$. ◀

▶ **Proposition 5.** *For all metric spaces (X, d) and all $\alpha < 1$, there is no centers algorithm that is α-fair in (X, d).*

Proof. Suppose the number of available centers is the same as the size of the population: $k = n$. Then $NR_{P,k}(x) = 0$ for all x, so $\alpha_{P,k}(A(P,k))$ is 1 if A places a center at every point in P and ∞ otherwise. ◀

▶ **Proposition 6.** *There exists a metric space (X, d) such that for all $\alpha < 2$, there is no centers algorithm that is α-fair in (X, d).*



🟨 **Figure 5** Under a graph metric, it is impossible to do better than 2-fairness when choosing four centers from among this set of points.

Proof. Consider the example in Figure 5, with 12 points under a graph metric with $k = 4$ and $P = X$. For every point $i \in P$, we have $NR(i) = 1$. But for any choice of three centers, some square will have at most one center, and one point in that square will therefore have travel distance at least 2. Thus, no centers algorithm in this metric space can be α-fair for any $\alpha < 2$. ◀

▶ **Proposition 7.** *For all $m \geq 2$ and all $\alpha < \sqrt{2}$, there is no centers algorithm that α-fair in m-dimensional Euclidean space.*

Proof. This holds by essentially the same example used to prove Proposition 6: P consists of 12 points arranged in three squares of unit side, where the distance between the squares is greater than the diameter of the squares, and $k = 4$. Once again, every point has neighborhood radius 1, and for any solution S, some square will have at most one center. Some other point i in that square will have travel distance $d(i, S) \geq \sqrt{2}$, and it follows immediately that $\alpha(S) \geq \sqrt{2}$. ◀