Clustering with Neighborhoods

Hongyao Huang
Department of Computer Science, University of Texas at Dallas, Richardson, TX, USA

Georgiy Klimenko
Department of Computer Science, University of Texas at Dallas, Richardson, TX, USA

Benjamin Raichel
Department of Computer Science, University of Texas at Dallas, Richardson, TX, USA

Abstract

In the standard planar k-center clustering problem, one is given a set P of n points in the plane, and the goal is to select k center points, so as to minimize the maximum distance over points in P to their nearest center. Here we initiate the systematic study of the clustering with neighborhoods problem, which generalizes the k-center problem to allow the covered objects to be a set of general disjoint convex objects $\mathcal{C}$ rather than just a point set P. For this problem we first show that there is a PTAS for approximating the number of centers. Specifically, if r_{opt} is the optimal radius for k centers, then in $n^{O(1/\varepsilon^2)}$ time we can produce a set of $(1+\varepsilon)k$ centers with radius $\leq r_{opt}$. If instead one considers the standard goal of approximating the optimal clustering radius, while keeping k as a hard constraint, we show that the radius cannot be approximated within any factor in polynomial time unless $\mathsf{P} = \mathsf{NP}$, even when $\mathcal{C}$ is a set of line segments. When $\mathcal{C}$ is a set of unit disks we show the problem is hard to approximate within a factor of $\frac{\sqrt{13}-\sqrt{3}}{2-\sqrt{3}} \approx 6.99$. This hardness result complements our main result, where we show that when the objects are disks, of possibly differing radii, there is a $(5+2\sqrt{3}) \approx 8.46$ approximation algorithm. Additionally, for unit disks we give an $O(n \log k) + (k/\varepsilon)^{O(k)}$ time $(1+\epsilon)$-approximation to the optimal radius, that is, an FPTAS for constant k whose running time depends only linearly on n. Finally, we show that the one dimensional version of the problem, even when intersections are allowed, can be solved exactly in $O(n \log n)$ time.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Clustering, Approximation, Hardness

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2021.6

Related Version *Full Version*: `https://arxiv.org/abs/2109.13302` [22]

Funding Partially supported by NSF CAREER Award 1750780.

1 Introduction

In the standard k-center clustering problem, one is given a set P of n points in a metric space and an integer parameter $k \geq 0$, and the goal is to select k points from the metric space (or from P in the discrete k-center problem), called centers, so as to minimize the maximum distance over points in P to their nearest center. Equivalently, the problem can be viewed as covering P with k balls with the same radius r, where the goal is to minimize r. It is well known that it is NP-hard to approximate the optimal k-center radius r_{opt} within any factor less than 2 in general metric spaces [21], and that the problem remains hard to approximate within a factor of roughly 1.82 in the plane [12]. For general metric spaces, the standard greedy algorithm of Gonzalez [19], which repeatedly selects the next center to be the point from P which is furthest from the current set of centers, achieves an optimal 2-approximation to r_{opt}. An alternative algorithm due to Hochbaum and Shmoys [20] also achieves an optimal approximation ratio of 2 by approximately searching for the optimal radius, observing that if $r \geq r_{opt}$ then all points will be covered after k rounds of repeatedly removing points in $2r$ radius balls centered at any remaining point of P.


© Hongyao Huang, Georgiy Klimenko, and Benjamin Raichel;
licensed under Creative Commons License CC-BY 4.0

In this paper we consider a natural generalization of k-center clustering in the plane, where the objects which we must cover are general disjoint convex objects rather than points. Specifically, in the *clustering with neighborhoods* problem the goal is to select k center points so that balls centered at these points with minimum possible radius intersect all the convex objects. This generalization is natural as real world objects may not be well modeled as individual points. This generalized setting has previously been considered for other classical point based problems in the plane, such as the Traveling Salesperson Problem [10], where the authors referred to these objects as neighborhoods. (We instead typically refer to them as *objects*.) To the best of our knowledge we are the first to consider the general problem of clustering convex objects in this context, though as we discuss below many closely related problems have been considered, some of which equate to special or extreme cases of our problem. We remark that since a point is a convex set, the hardness results for k-center clustering immediately apply to clustering with neighborhoods.

Related Work

As clustering is a fundamental data analysis task, countless variants have been considered. Here we focus on variants which share our k-center objective of minimizing the maximum radius of the balls at the chosen centers. Bandyapadhyay et al. [7] considered the colorful k-center problem, where the points are partitioned into color classes $P_1, \ldots, P_c$ and the goal is to find k balls with minimum radius which cover at least t_i points from each color class P_i. When our convex objects have bounded diameter our problem can be approximately cast as an instance of colorful k-center by replacing each object with the set P_i of grid points it intersects and setting $t_i = 1$. General colorful clustering, however, is more challenging as the color classes can be interspersed, which is why [7] assumes the number of color classes is a constant, allowing for a constant factor approximation, which subsequently was improved [5, 23]. Note that colorful k-center itself generalizes the k-center with outliers problem [8], corresponding to the case with a single color class P with $n - t$ outliers allowed.

Xu and Xu [31] considered k-center clustering on point sets (KCS) where given points sets $S_1, \ldots, S_n$ the goal is to find k balls of minimum radius such that each S_i is entirely contained in one of the balls. Again when our objects have bounded diameter we can relate our problem to KCS by discretizing the objects. Their requirement that all of S_i be covered by a single ball implies that the optimal radius is at least the radius of the largest object, whereas in our case as only a single point of S_i needs to be covered the radius can be arbitrarily smaller. In particular, while [31] achieves a $(1 + \sqrt{3})$-approximation, we show our problem cannot in general be approximated within any factor in polynomial time unless $\mathsf{P} = \mathsf{NP}$.

For the special case when $k = 1$ or $k = 2$, there are several prior results which closely relate to our problem. When $k = 1$, i.e. the one-center problem, the solution can be derived from the farthest object Voronoi diagram, for which Cheong et al. [9] gave a near linear time algorithm for polygon objects. For disk objects, Ahn et al. [4] gave a near quadratic time algorithm for the two-center problem. Several papers have also considered generalizing to higher dimensions, but restricting the convex objects to affine subspaces of dimension Δ. Gao et al. [16] introduced the 1-center problem for n lines, achieving a linear time $(1 + \varepsilon)$-approximation, as well as a $(1 + \varepsilon)$-approximation for higher dimensional flats or convex sets whose running time depends exponentially on Δ. Later in [17] the same authors considered the more challenging $k = 2$ and $k = 3$ cases for lines, providing a $(2 + \varepsilon)$-approximation in quasi-linear time. Subsequently, [26] considered the problem for axis-parallel flats, where they provide an improved approximation for $k = 1$, hardness results for $k = 2$, and an

approximation for larger k where the time depends exponentially on both k and Δ. While our focus is on the k-center objective, we remark that k-means clustering for lines was considered by Marom and Feldman [27], who gave a PTAS for constant k.

The k-center problem for points in a metric space can also be viewed as clustering the vertices according to the shortest path metric of a positively weighted graph. This allows one to consider specific graph classes, for example, Eisenstat et al. [11] gave a polynomial time bi-criteria approximation scheme for k-center in planar graphs (i.e. they allow both the number of centers and radius to be violated). We remark, however, that for our problem, and the various others described above where the objects are not points, the complete graph with all pairwise distances between the objects, is not necessarily metric (i.e. it may not be its own metric completion). For example, the triangle inequality would be violated if you had two small convex objects (e.g. points) which are far from one another but both are close to some other large convex object. Note that this non-metric behavior is what allows us to prove a stronger hardness of approximation result than that for points in the plane [12].

Finally, we note that there is a polynomial time algorithm for k-center when k is a constant and the objects are points in d-dimensional Euclidean space, for constant d. Specifically, Agarwal and Procopiuc [3] gave an $n^{O(k^{1-1/d})}$ time exact algorithm, as well as a $O(n \log k) + (k/\varepsilon)^{O(k^{1-1/d})}$ time $(1+\varepsilon)$-approximation. Later Bădoiu et al. [6] removed the bounded dimension assumption, achieving a $2^{O((k \log k)/\varepsilon^2)} \cdot dn$ time $(1+\varepsilon)$-approximation.

Our Contribution

In this paper we initiate the systematic study of the NP-hard clustering with neighborhoods problem. While this problem allows centers to be placed anywhere in the plane, in Section 3 we first argue that one can compute a cubic sized set of points P and a cubic sized set of radii R, such that for any integer $k \geq 0$ there is an optimal set $S \subseteq P$ of k centers with optimal radius $r_{opt} \in R$. This naturally leads to a PTAS for approximating the optimal number of centers by using Minkowski sums to reduce the problem to instances of geometric hitting set, for which there is a well known PTAS [29]. Specifically, if r_{opt} is the optimal radius for k centers, then in $n^{O(1/\varepsilon^2)}$ time we can produce a set of $(1+\varepsilon)k$ centers with radius $\leq r_{opt}$.

In clustering problems, however, often the emphasis is on approximating the radius, while keeping k as a hard constraint. In Section 4 we prove this problem is significantly harder, by adapting the hardness proof of [12] for planar k-center. Specifically, we show that the radius cannot be approximated within any factor in polynomial time unless P = NP, even when the convex objects are restricted to disjoint line segments. On the other hand, for disjoint unit disks, a more in depth proof shows the problem is APX-hard, and in particular cannot be approximated within $\frac{\sqrt{13}-\sqrt{3}}{2-\sqrt{3}} \approx 6.99$ in polynomial time unless P = NP. Complementing this result, in Section 5 we present our main result, showing that when the objects are disjoint disks (of possibly varying radii) there is a $(5+2\sqrt{3})$-approximation for the optimal radius. Significantly, for the case of disks, our approximation factor of $5+2\sqrt{3} \approx 8.46$ is close to our hardness bound of $\frac{\sqrt{13}-\sqrt{3}}{2-\sqrt{3}} \approx 6.99$. Moreover, while our approximation holds for disks of varying radii, interestingly our hardness bound applies even for disks of uniform radii.

Further probing the complexity of clustering with neighborhoods, in Section 6 we show there is an FPTAS for unit disks when k is bounded by a constant. Specifically, we give an $O(n \log k) + (k/\varepsilon)^{O(k)}$ time $(1+\varepsilon)$-approximation to the optimal radius, by carefully reducing to the algorithm of [3] for k-center. Finally in Section 7, by utilizing the searching procedure of [15], we show that in one dimension the problem can be solved exactly in $O(n \log n)$ time even when intersections are allowed, contrasting our hardness results in the plane.

2 Preliminaries

Given points $x, y \in \mathbb{R}^d$, $||x - y||$ denotes their Euclidean distance. Given two closed sets $X, Y \subset \mathbb{R}^d$, $||X - Y|| = \min_{x \in X, y \in Y} ||x - y||$ denotes their distance. For a single point x we write $||x - Y|| = ||\{x\} - Y||$. For a point x and a value $r \geq 0$, let $B(x, r)$ denote the closed ball centered at x and with radius r.

Let $\mathcal{C}$ be a set of n pairwise disjoint convex objects in the plane. For simplicity, we assume $\mathcal{C}$ is in general position. We work under the standard assumption that the objects in $\mathcal{C}$ are semi-algebraic sets of constant descriptive complexity. Namely, the boundary of each object is composed of a set of algebraic arcs where the sum of the degrees of these arcs is bounded by a constant, and any natural standard operation on such objects, such as computing the distance between any pair of objects, can be carried out in constant time. See Agarwal et al. [1] for a more detailed discussion of this model. Our analysis generalizes to the case where n is the total complexity of $\mathcal{C}$ and individual objects in $\mathcal{C}$ are not required to have constant complexity, however, assuming constant complexity simplifies certain structural statements and the polynomial degree of n in our running time statements.

▶ **Problem 1** (Clustering with Neighborhoods)**.** *Given a set $\mathcal{C}$ of n disjoint convex objects in the plane, and an integer parameter $k \geq 0$, find a set of k points S (called centers) which minimize the maximum distance to a convex object in $\mathcal{C}$. That is,*

$$S = \arg \min_{S' \subset \mathbb{R}^2, |S'| = k} \max_{C \in \mathcal{C}} ||C - S'||.$$

Let S be any set of k points, and let $r = \max_{C \in \mathcal{C}} ||C - S||$. We refer to r as the *radius* of the solution S, since r is the minimum radius such that the set of all balls $B(s, r)$ for $s \in S$, intersect all $C \in \mathcal{C}$. If S is an optimal solution then we refer to its radius r_{opt} as the optimal radius.

In this paper we will consider two types of approximations.[1] Let $\mathcal{C}, k$ be an instance of Problem 1 with optimal radius r_{opt}. For a value $\alpha \geq 1$, we refer to a polynomial time algorithm as an *α-size-approximation* if it returns a solution S of radius $\leq r_{opt}$ where $|S| \leq \alpha k$. Alternatively, we refer to a polynomial time algorithm as an *α-radius-approximation* if it returns a solution S of radius $\leq \alpha r_{opt}$ where $|S| = k$. Often we refer to the latter radius case simply as an α-approximation.

3 Canonical Sets and a PTAS for Approximating the Size

In this section we show that while Problem 1 allows centers to be placed anywhere in the plane, we can compute a canonical cubic sized set of points P and a set of corresponding radii R, such that for any integer $k \geq 0$ there is an optimal set $S \subseteq P$ of k centers with optimal radius $r_{opt} \in R$. We then use this property to give a PTAS for Problem 1 when approximating the size of an optimal solution. Specifically, for any fixed $\varepsilon > 0$, we give a $(1 + \varepsilon)$-size-approximation with running time $n^{O(1/\varepsilon^2)}$. In Section 5, we will again use this canonical set when designing our constant factor radius-approximation for disks.

The *bisector* of two convex objects C, C' is the set of all points x in the plane such that $||x - C|| = ||x - C'||$. Let $\beta(C, C')$ denote the bisector of C and C'. As discussed in [24], any set $\mathcal{C}$ of n disjoint constant-complexity convex objects in general position satisfies the conditions of an abstract Voronoi diagram [25]. In particular we can assume the following:

[1] We refrain from using the standard bi-criteria approximation terminology to emphasize that in each case only the size or only the radius is being approximated, not both.

1) For any $C, C' \in \mathcal{C}$ we have that $\beta(C, C')$ is an unbounded simple curve.
2) The intersection of any two bisectors is a discrete set with a constant number of points.
We point out that in the following lemma there is a single pair of sets P, R which works simultaneously for all values of k.

▶ **Lemma 2.** *Let $\mathcal{C}$ be a set of n disjoint convex objects. In $O(n^3 \log n)$ time one can compute a set of $O(n^3)$ points P, and a corresponding set of $O(n^3)$ radii R, such that for any value $k \geq 0$ for the instance $\mathcal{C}, k$ of Problem 1 there is an optimal set of k centers S with optimal radius r_{opt} such that $S \subseteq P$ and $r_{opt} \in R$.*

Proof. Let $I(\mathcal{C})$ be a set containing exactly one (arbitrary) point from each convex object in $\mathcal{C}$. For any number of centers k, let S be any optimal solution, and let r_{opt} be the optimal radius. Consider an arbitrary center $s \in S$. Let $\mathcal{C}'$ be the subset of objects in $\mathcal{C}$ which intersect the ball $B(s, r_{opt})$. We can assume $\mathcal{C}'$ is non-empty, as otherwise the center s does not cover any convex object within radius r_{opt} and so can be thrown out. Moreover, if $|\mathcal{C}'| = 1$ then we can assume s is the point from $I(\mathcal{C})$ which intersects this one convex object. So assume $|\mathcal{C}'| > 1$, and let C be the convex object in $\mathcal{C}'$ which lies furthest from s. Now consider moving s continuously toward the convex object C. As we do so the distance from s to C monotonically decreases. Thus so long as C remains the furthest convex object from s in $\mathcal{C}'$, the ball $B(s, r_{opt})$ still intersects all of $\mathcal{C}'$ (i.e. we did not increase the solution radius). Now if C always remains the furthest, when s eventually reaches and intersects C then this will imply its distance to all objects in $\mathcal{C}'$ is zero, which is a contradiction as we assumed the convex objects do not intersect. Otherwise, at some point C is no longer the furthest, which implies we must have crossed a bisector $\beta(C, C')$ for some other convex object $C' \in \mathcal{C}'$.

So far we have shown one can assume each center s either is in $I(\mathcal{C})$, or lies on the bisector $\beta = \beta(C, C')$ of the two objects, C, C', which lie furthest away from s among the set of objects $\mathcal{C}'$ which intersect the ball $B(s, r_{opt})$. In the latter case, let T_β denote the set of all points p on β such that there exists a third object $X \in \mathcal{C}$ such that $||p - X|| = ||p - C||$ (or equivalently $||p - X|| = ||p - C'||$). Note that such points lie at intersections of bisectors and thus from the above discussion before the lemma, we know T_β is a discrete set. As β is a simple curve, we can view points in T_β as being ordered along β. Suppose that $s \notin T_\beta$, and let p and q be the points of T_β which come immediately before and after s along β, and let $[p, q]$ denote the portion of β lying between these points. (This interval may be unbounded to one side if s comes after or before all points in T_β.) Recall that $\mathcal{C}'$ is the subset of objects intersecting $B(s, r_{opt})$, and C and C' are the furthest from s among those in $\mathcal{C}'$. Observe that for any other point z in $[p, q]$, C and C' must also be the furthest objects from z among those in $\mathcal{C}'$, as otherwise as we move continuously along β from s to z we must cross another point from T_β before reaching z and there are no such points in (p, q). Thus if we replace s with the point in $[p, q]$ minimizing the distance to C (or equivalently C') then all objects previously intersected by $B(s, r_{opt})$ will remain intersected by $B(s, r_{opt})$.

Let $M(\mathcal{C})$ be a set containing, for each bisector β, the set T_β and one minimum distance point from each such interval $[p, q]$. We thus have argued that the points of S can be assumed to lie in $P = I(\mathcal{C}) \cup M(\mathcal{C})$. As for the running time and size of these sets, first observe that $I(\mathcal{C})$ has size n and can be trivially computed in $O(n)$ time. For the set $M(\mathcal{C})$, first observe that there are $O(n^2)$ bisectors. For any bisector β, the set T_β of intersection points of β with other bisectors that are equidistant at the intersection point, has size $O(n)$, since by general position every point is equidistant to at most 3 objects and as mentioned above any pair of bisectors intersect in a constant number of points. (In other words, we ultimately consider all $O(n^3)$ points equidistant to three objects, as opposed to all $O(n^4)$ bisector intersections.) Thus the set $M(\mathcal{C})$, and correspondingly P, has size $O(n^3)$ as claimed. For the running

time, as the objects in $\mathcal{C}$ all have constant complexity, so do their bisectors, and thus T_β can be computed in $O(n)$ time. The minimum points of $M(\mathcal{C})$ on β, can thus be computed by sorting T_β along β, in $O(n \log n)$ time, and then computing the minimum point in constant time for each constant complexity interval between consecutive pairs of points from T_β along β. Thus over all $O(n^2)$ bisectors it takes $O(n^3 \log n)$ time to compute $M(\mathcal{C})$. ◀

We now argue the canonical sets P and R from the above lemma naturally lead to a PTAS for size-approximation by using Minkowski sums. For sets $A, B \subset \mathbb{R}^2$, let $A \oplus B = \{a + b \mid a \in A,\ b \in B\}$ denote their Minkowski sum. Let $B(r)$ denote the ball of radius r centered at the origin. Then we write $\mathcal{C} \oplus B(r) = \{C \oplus B(r) \mid C \in \mathcal{C}\}$. A set of points S is called a *hitting set* for a set of objects if every object has non-empty intersection with S.

▶ **Observation 3.** *A set S of k centers is a solution to Problem 1 of radius r if and only if S is a hitting set of size k for $\mathcal{C} \oplus B(r)$. This holds since for any $C \in \mathcal{C}$ and $s \in S$, $B(s, r) \cap C \neq \emptyset$ if and only if $s \in C \oplus B(r)$.*

In the geometric hitting set problem we are given a set $\mathcal{R}$ of n regions and a set P of m points in the plane, and the goal is to select a minimum sized hitting set for $\mathcal{R}$ using points from P. The above observation implies we can reduce any given instance $\mathcal{C}, k$ of Problem 1 to multiple instances of geometric hitting set. Specifically, by Lemma 2, in $O(n^3 \log n)$ time we can compute a set R of $O(n^3)$ values, one of which must be the optimal radius r_{opt}. Then for each $r \in R$ we construct a hitting set instance where $\mathcal{R} = \mathcal{C} \oplus B(r)$, and P is the set of points from Lemma 2. By the above observation, if $r < r_{opt}$, then the hitting set instance requires more than k points, and if $r \geq r_{opt}$ then it requires at most k points. Therefore, given an algorithm for geometric hitting set we can use it to binary search for r_{opt}.

While hitting set is in general NP-hard to approximate within logarithmic factors [30], in our case there is a PTAS as the regions are nicely behaved. A collection of regions in the plane is called a set of *pseudo-disks* if the boundaries of any two distinct regions in the set cross at most twice. Mustafa and Ray [29] showed that there is an $nm^{O(1/\varepsilon^2)}$ time PTAS for geometric hitting set when $\mathcal{R}$ is a collection of n pseudo-disks and P is a set of m points. It is known that if we take the Minkowski sum of a single convex object with each member of a set of disjoint convex objects, then the resulting set is a collection of pseudo-disks (see for example [2]). Thus $\mathcal{C} \oplus B(r)$ is a collection of pseudo-disks. Therefore, by the above discussion, we have the following theorem. As the decision procedure is now approximate, the binary search must be modified to look at larger radii when the hitting set algorithm returns $> (1 + \varepsilon)k$ points, and smaller radii otherwise. (This yields an adjacent pair $r < r'$ such that $r < r_{opt}$, implying $r' \leq r_{opt}$, and an r'- cover of the input using $\leq (1 + \varepsilon)k$ points.)

▶ **Theorem 4.** *There is a PTAS for Problem 1 for approximating the optimal solution size. That is, for any fixed $\varepsilon > 0$, there is a $(1 + \varepsilon)$-size-approximation with running time $n^{O(1/\varepsilon^2)}$.*

We remark that the PTAS of [29] implicitly assumes the objects are in general position, that is if two objects intersect then they properly intersect (i.e. their interiors intersect). While $\mathcal{C}$ satisfies this property, it may not after we take the Minkowski sum with a given radius. However, as we can compute distances between our objects, this is easily overcome by computing the smallest non-zero distance d between two objects in $C \oplus B(r)$, and instead running the hitting set algorithm on $C \oplus B(r + \alpha)$, where α is some infinitesimal value less than $d/2$. This ensures any objects which intersected in $C \oplus B(r)$ now properly intersect, and there are no new intersections.

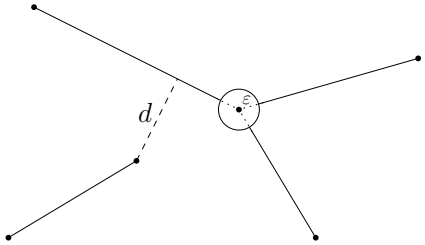

Figure 4.1 Reducing planar Vertex Cover to Problem 1 for segments.

4 Radius Approximation Hardness

In this section we argue that for Problem 1 it is hard to approximate the radius within any factor, even when $\mathcal{C}$ is restricted to being a set of line segments. Moreover, for the case when $\mathcal{C}$ is a set of disks, i.e. the case considered in Section 5, we argue the problem is APX-Hard. Our hardness results use a construction similar to the one from [12], where they reduce from the problem of planar vertex cover where the maximum degree of a vertex is three, which is known to be NP-complete [18]. We denote this problem as P3VC.

4.1 Line Segments

Here we argue that it is hard to radius-approximate Problem 1 within any factor, even when $\mathcal{C}$ is a set of line segments. We remark that the following reduction works for any instance of planar vertex cover (i.e. regardless of the degree), but the reduction for disks in the next subsection uses that the degree is at most three.

▶ **Theorem 5.** *Problem 1 cannot in polynomial time be radius-approximated within any factor that is computable in polynomial time unless* P = NP*, even when restricting to the set of instances in which $\mathcal{C}$ is a set of disjoint line segments.*

Proof. Let G, k be an instance of P3VC. Consider a straight line embedding of G, and let d denote the distance between the closest pair of non-adjacent segment edges.[2] Let $\varepsilon > 0$ be a value strictly smaller than $d/2$ and strictly smaller than half the length of any segment edge. The set $\mathcal{C}$ of segments in our instance of Problem 1 will be the segment edges from the embedding, but where each segment has an ε amount removed from each end, i.e. we remove all portions of segments in ε balls around the vertices, see Figure 4.1. We use the same value of k in our Problem 1 instance as in the P3VC instance.

If there is a vertex cover of size at most k then if we place balls of radius ε at each of the k corresponding vertices of the embedding, then these balls will intersect all segments in $\mathcal{C}$, i.e. we have a solution to Problem 1 of radius ε. On the other hand, by the definition of d, any ball of radius $< d/2$ cannot simultaneously intersect two segments from $\mathcal{C}$ if they correspond to non-adjacent edges from G. (Note when we shrunk the edges by ε this could only have made them further apart.) Thus if the minimum vertex cover requires $> k$ vertices, then our instance of Problem 1 requires $> k$ centers if we limit to balls with radius $< d/2$.

[2] In $O(n \log n)$ time one can compute a straight line embedding of G where the vertices are on an $(2n-4) \times (n-2)$ grid [14]. This implies a lower bound on d with a polynomial number of bits.

Therefore, if we could approximate the minimum radius of our Problem 1 instance within any factor less than $d/(2\varepsilon)$ then we can determine whether the corresponding vertex cover instance had a solution with $\leq k$ vertices. However, we are free to make $\varepsilon > 0$ as small as we want and thus $d/(2\varepsilon)$ as large as we want, so long as this quantity (or more precisely a lower bound on it) is computable in polynomial time. ◀

4.2 Disks

Here we argue that it is hard to radius-approximate Problem 1 within a constant factor when $\mathcal{C}$ is restricted to be a set of unit disks. The following reduction from P3VC is similar to the one given in [12], which embeds the graph such that edges are replaced by odd length sequences of points. In our case, these odd length sequences of points are instead replaced with odd length sequences of appropriately spaced disks.

▶ **Theorem 6.** *For the set of instances in which $\mathcal{C}$ is a set of disjoint unit disks, Problem 1 cannot be radius-approximated to any factor less than $\frac{\sqrt{13}-\sqrt{3}}{2-\sqrt{3}}$ in polynomial time unless* P = NP.

Proof. To simplify our construction description, instead of requiring the disks be disjoint, we allow them to intersect at their boundaries, but not their interiors. Later we remark how this easily implies the result for the disjoint disk case.

So let $G = (V, E), k$ be an instance of P3VC. For every edge in E we create a sequence of an odd number (greater than 1) of unit disks, where consecutive disks in the sequence are spaced $2(2/\sqrt{3} - 1)$ apart from one another. (Note $2(2/\sqrt{3} - 1)$ is the distance between the disks, not their centers.) For a vertex v of degree two, we place the disks corresponding to the v end of the adjacent edges again at distance $2(2/\sqrt{3} - 1)$ apart. For a vertex v of degree three, we place the disks corresponding to the v end of the adjacent edges such that they all just touch one another at their boundaries, see Figure 4.3. Thus the centers of these disks form an equilateral triangle, and let the center point of this triangle be t. For any one of the adjacent edges, we further require that the centers of the first two disks (on the v end of the edge) lie on a straight line containing t, in other words the edges leaving v do not bend until several disks away from v. As G is a planar graph with maximum degree three such an embedding of polynomial size is possible, similar to the case in [12]. Doing so requires using different numbers of disks for each edge and allowing the edges to bend (i.e. the centers of three consecutive disks of an edge may not lie on a line). However, we will require these bends to be gradual. Specifically, observe that if the centers of three consecutive disks of an edge were on a straight line, the distance between the two non-consecutive disks would be $2(1 + 2(2/\sqrt{3} - 1)) > 2.6$, see Figure 4.2. We then require that the bends are shallow enough such that two non-consecutive disks of an edge are more than 2.5 apart. We also require this for disks from edges adjacent to a degree two vertex (when they are not both the disks immediately adjacent at the vertex), or a degree three vertex when neither disk is one of corresponding three touching disks of the vertex. Finally, for disks that come from edges that are not adjacent, we easily enforce that they are again more than 2.5 apart. (This is similar to the value d from Theorem 5.)

So given an instance G, k of P3VC, we construct an instance $\mathcal{C}, \kappa$ of Problem 1 where $\mathcal{C}$ is determined from G as described above and $\kappa = k + (|\mathcal{C}| - |E|)/2$. We first argue if G has a vertex cover of size k then for our instance of Problem 1 there is a solution of radius $2/\sqrt{3} - 1$. First, for any vertex v in the vertex cover we create a center, and roughly speaking place it at the location of v in the embedding. Namely, if v had degree two then we place the center at the midpoint of the centers of the disks at the ends of the edges adjacent to v, which

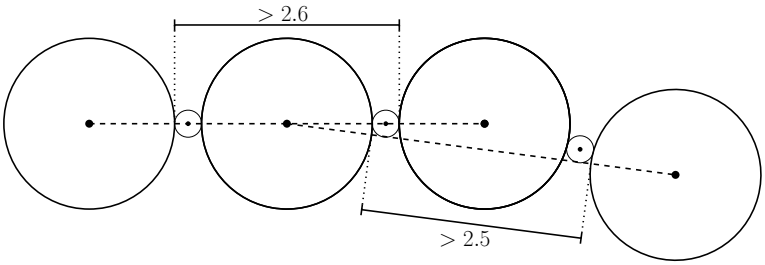

Figure 4.2 Consecutive disks along an edge.

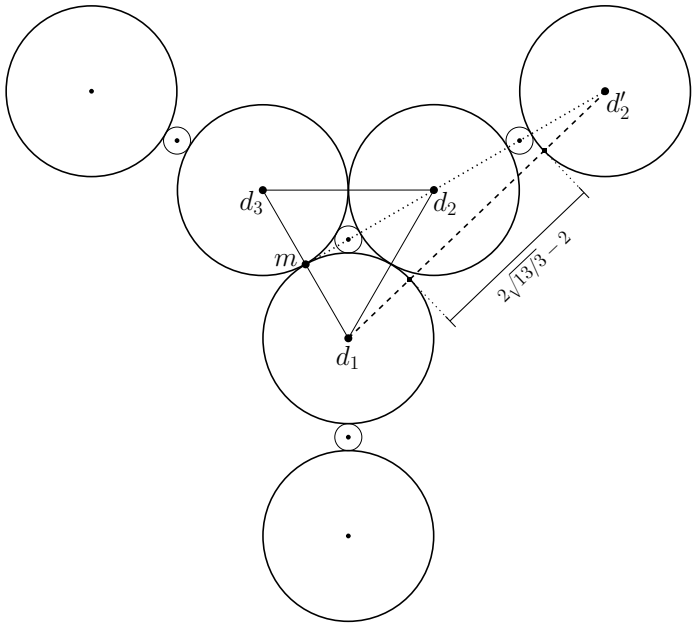

Figure 4.3 Three touching disks corresponding to a degree three vertex.

by construction are exactly $2(2/\sqrt{3}-1)$ apart and thus a ball at the midpoint with radius $2/\sqrt{3}-1$ intersects both. If v has degree three then we place a center at the center point t of the equilateral triangle determined by three touching disks of the adjacent edges. An easy calculation[3] shows that since our disks have unit radius, that $B(t, 2/\sqrt{3}-1)$ intersects the three touching disks. We now cover the remaining disks with $(|\mathcal{C}|-|E|)/2$ centers. For any edge $e \in E$ let n_e be the number of disks used for e in the above construction. Observe that as we already placed centers at vertices corresponding to a vertex cover of the edges, at least one disk at the end of each edge is already covered, and so there are at most $n_e - 1$ consecutive disks that need to be covered. (Note $n_e - 1$ is even.) However, as consecutive disks are exactly $2(2/\sqrt{3}-1)$ apart on each edge, these $n_e - 1$ disks can be covered with $(n_e-1)/2$ balls of radius $(2/\sqrt{3}-1)$ by covering the disks in pairs. Thus the total number of centers used is $k + \sum_{e\in E}(n_e-1)/2 = k + (|\mathcal{C}|-|E|)/2 = \kappa$.

Now suppose the minimum vertex cover of G requires $> k$ vertices. In this case we argue that our instance of Problem 1 requires more than κ centers if we limit to balls with radius $< \sqrt{13/3}-1$. Call any two disks in $\mathcal{C}$ neighboring if they are consecutive on an

[3] For an equilateral triangle with edge length 2, the distance from an edge to the center point of the triangle is $1/\sqrt{3}$, thus the distance from the center point to any one of the unit balls is $2/\sqrt{3}-1$.

edge or if they are disks on the v end of two edges adjacent to a vertex v. By construction, neighboring disks have distance $\leq 2/\sqrt{3} - 1$ from each other. For a pair of disks which are not neighboring we now argue their distance is at least $2\sqrt{13/3} - 2$. Specifically, if these disks come from the same edge but are not consecutive along that edge, or if they are from distinct edges that are either non-adjacent or are adjacent to a degree two vertex (but not the two disks of that vertex), then by construction their distance is $> 2.5 > 2\sqrt{13/3} - 2$. The remaining case is when the disks are from distinct edges adjacent to a degree three vertex, but they are not both from the three touching disks of the vertex. It is easy to see that the closest two such disks can be is when one of the disks is one of the three touching disks, and the other is the second disk on another edge. We now calculate the distance between two such disks, see Figure 4.3. Let the three touching disks be denoted D_1, D_2, and D_3, with centers d_1, d_2, and d_3, respectively. Let D_2' denote the second disk on the edge containing D_2, and let its center by d_2'. We wish to compute $||D_1 - D_2'|| = ||d_1 - d_2'|| - 2$, as these are unit disks. Let m denote the midpoint of d_1 and d_3, and observe that the line through d_2 and d_2' passes through m and is orthogonal to the line through d_1 and d_3, as the points d_1, d_2, and d_3 form and equilateral triangle. Thus by the Pythagorean theorem we have $||d_1 - d_2'||^2 = 1^2 + (1 + 2(2/\sqrt{3} - 1) + 1 + \sqrt{3})^2 = 1 + (4/\sqrt{3} + \sqrt{3})^2 = 52/3$, where the $+\sqrt{3}$ term is the height of an equilateral triangle of side length 2. Thus $||D_1 - D_2'|| = ||d_1 - d_2'|| - 2 = 2\sqrt{13/3} - 2$.

Now we finish the argument that when the minimum vertex cover of G requires $> k$ vertices, our instance of Problem 1 requires more than κ centers if we limit to balls with radius $< \sqrt{13/3} - 1$. By the above, limiting to radius $< \sqrt{13/3} - 1$ implies that any ball either covers just a single disk, or a pair of neighboring disks. An edge e with n_e disks thus requires at least $\lceil n_e/2 \rceil = 1 + (n_e - 1)/2$ disks to cover it. Moreover, a ball can only cover both a disk of e and e' if those disks are on the v end of two edges adjacent to v. Let E_z be the subset of edges with at least one disk covered by such a ball (i.e. a ball corresponding to a vertex), and let z be the number of such balls. Then the total number of balls required is

$$\begin{aligned}
&\geq z + \sum_{e \in E_z} (n_e - 1)/2 + \sum_{e \in E \setminus E_z} (1 + (n_e - 1)/2) \\
&= z + (|\mathcal{C}| - |E|)/2 + |E \setminus E_z| = z + (\kappa - k) + |E \setminus E_z|,
\end{aligned}$$

which is more than κ when $z + |E \setminus E_z| > k$. Notice, however, there is a vertex cover of G of size $z + |E \setminus E_z|$, consisting of the vertices that z counted, and one vertex from either end of each edge in $E \setminus E_z$. Thus as the minimum vertex cover has size $> k$, we have $z + |E \setminus E_z| > k$ as desired.

Therefore, if we could approximate the minimum radius of our Problem 1 instance within any factor less than $\frac{\sqrt{13/3}-1}{2/\sqrt{3}-1} = \frac{\sqrt{13}-\sqrt{3}}{2-\sqrt{3}}$ then we can determine whether the corresponding vertex cover instance had a solution with $\leq k$ vertices. In the above analysis the boundaries of the circles were allowed to intersect, but we can enforce that all disks are disjoint without changing the approximation hardness factor since we showed the problem is hard for any factor that is less than $\frac{\sqrt{13}-\sqrt{3}}{2-\sqrt{3}}$. Specifically, rather than having the disks for a degree three vertex touch, we can instead make them arbitrarily close to touching. ◀

5 Constant Factor Radius Approximation for Disks

In this section we argue that when $\mathcal{C}$ is a set of disjoint disks (of possibly differing radii), that there is a constant factor radius-approximation for Problem 1.

▶ **Lemma 7.** *Let $\mathcal{C}$ be a set of pairwise disjoint disks such that for all $C \in \mathcal{C}$, the radius of C is $\geq r$. If there is a point $s \in \mathbb{R}^2$ where $||s - C|| \leq (2/\sqrt{3} - 1)r$ for all $C \in \mathcal{C}$, then $|\mathcal{C}| \leq 2$.*

Proof. We give a proof by contradiction. So suppose there exists a point s such that there are three disjoint disks in $\mathcal{C}$, each with radius $\geq r$, and all of which intersect the ball $B(s, (\frac{2}{\sqrt{3}} - 1)r)$. Observe that if any one of these three disjoint disks C has radius $> r$, then it can be replaced by a disk C' of radius r such that $C' \subset C$ and C' still intersects $B(s, (\frac{2}{\sqrt{3}} - 1)r)$. As these new disks are all still disjoint and intersect $B(s, (\frac{2}{\sqrt{3}} - 1)r)$, it suffices to argue we get a contradiction when all three disks have radius exactly r. Let the centers of these three disks be denoted x, y, and z. Now, at least one of the angles $\angle xsy$, $\angle ysz$, and $\angle zsx$ is $\leq 2\pi/3$. Without loss of generality assume it is $\angle xsy$, and let $\gamma = \angle xsy$.

Consider the triangle $\triangle sxy$, and let its side lengths be denoted $a = ||x-s||, b = ||y-s||, c = ||x - y||$. Since $\gamma \leq 2\pi/3$, by the Law of Cosines we thus have $c^2 = a^2 + b^2 - 2ab\cos(\gamma) \leq a^2 + b^2 + ab$. As the r radius disks with centers x and y are disjoint, we know that $2r < c$. Combining these two inequalities we get $4r^2 < a^2 + b^2 + ab$. As $B(s, (\frac{2}{\sqrt{3}} - 1)r)$ intersects the r radius disks centered at both x and at y, we also have that $a, b \leq (\frac{2}{\sqrt{3}} - 1)r + r = \frac{2r}{\sqrt{3}}$. Combining this with the previous inequality gives $4r^2 < a^2+b^2+ab \leq 4r^2/3+4r^2/3+4r^2/3 = 4r^2$, which is a clear contradiction and thus the number of disks in $\mathcal{C}$ is at most 2. ◀

For any constant $c \geq 1$, we call an algorithm a *c-decider* for Problem 1, if for a given instance with optimal radius r_{opt}, and for any given query radius r, if $r \geq r_{opt}$ then the algorithm returns a solution S of radius $\leq cr$, and if $r < r_{opt}/c$ it returns False (for $r_{opt}/c \leq r < r_{opt}$ either answer is allowed).

▶ **Lemma 8.** *There is an $O(n^{2.5})$ time $(5 + 2\sqrt{3})$-decider for Problem 1, when restricted to instances where $\mathcal{C}$ is a set of disjoint disks.*

Proof. Let r be the given query radius. We build a set S of centers as follows, where initially $S = \emptyset$. Let P be the set of center points of all disks in $\mathcal{C}$ with radius $< (3 + 2\sqrt{3})r$. Until P is empty repeatedly add an arbitrary point $p \in P$ to the set S, remove all disks from $\mathcal{C}$ which intersect $B(p, (5 + 2\sqrt{3})r)$, and remove all center points from P corresponding to disks removed from $\mathcal{C}$. Let S_1 refer to the resulting set of centers. For the remaining set of disks $\mathcal{C}'$, define the subset $\mathcal{C}'' = \{C \in \mathcal{C}' \mid \exists\ D \in \mathcal{C}' \setminus \{C\} \text{ s.t. } ||C - D|| \leq 2r\}$. First, for every disk C in $\mathcal{C}' \setminus \mathcal{C}''$ we add an arbitrary point from C to S. Let this set of added centers be denoted S_2. Now for the set $\mathcal{C}''$ we construct a graph $G = (V, E)$ where $V = \mathcal{C}''$ and there is an edge from C to D if and only if $||C - D|| \leq 2r$. Let $\mathcal{E}$ be a minimum edge cover of G. (Note every vertex in G has an adjacent edge by the definition of $\mathcal{C}''$ and thus $\mathcal{E}$ exists.) For every edge $(C, D) \in \mathcal{E}$, $||C - D|| \leq 2r$ and thus there is a point $p \in \mathbb{R}^2$ such that $||p - C||, ||p - D|| \leq r$. So finally, for each $(C, D) \in \mathcal{E}$ we add this corresponding point p to S. Let this final set of added centers be denoted S_3. If $|S| \leq k$ we return S (which is the disjoint union of S_1, S_2, and S_3) and otherwise we return False.

To prove the above algorithm is a $(5+2\sqrt{3})$-decider, first we argue that if $r < r_{opt}/(5+2\sqrt{3})$ then it returns False. To do so we prove the contrapositive. So assume $|S| \leq k$. Let S_1, S_2, and S_3, and $\mathcal{C}'' \subseteq \mathcal{C}' \subseteq \mathcal{C}$ be as defined above. As we used balls of radius $(5 + 2\sqrt{3})r$, all $C \in \mathcal{C} \setminus \mathcal{C}'$ are within distance $(5 + 2\sqrt{3})r$ of points in S_1. All $C \in \mathcal{C}' \setminus \mathcal{C}''$ have distance zero to a point in S_2. Finally, all $C \in \mathcal{C}''$ have distance $\leq r$ to a point in S_3. As S is the disjoint union of S_1, S_2, and S_3, we thus have that all $C \in \mathcal{C}$ are within distance $(5 + 2\sqrt{3})r$ to a set S with $\leq k$ points, which by the definition of Problem 1 means that $r_{opt} \leq (5 + 2\sqrt{3})r$.

Now suppose $r \geq r_{opt}$, where r_{opt} is the optimal radius for the given instance $\mathcal{C}, k$ of Problem 1. In order to prove the algorithm is a $(5 + 2\sqrt{3})$-decider, in this case we must argue it returns a $\leq (5 + 2\sqrt{3})r$ radius solution. As already shown above, if the algorithm

returns a solution then it has radius $\leq (5+2\sqrt{3})r$, thus all we must argue is that a solution is returned, namely that $|S| \leq k$. So fix an optimal solution S^* for the original input instance $\mathcal{C}, k$. We argue that there are disjoint subsets S_1^*, S_2^*, and S_3^* of S^* such that $|S_1^*| \geq |S_1|$, $|S_2^*| \geq |S_2|$, and $|S_3^*| \geq |S_3|$, and therefore $|S| \leq |S^*| = k$.

Let the points in $S_1 = \{t_1, \ldots, t_{|S_1|}\}$ be indexed in the order they were selected. Consider the point t_i, which is the center of some disk $C_i \in \mathcal{C}$ with radius $\leq (3+2\sqrt{3})r$. Let $U_i = \cup_{j \leq i} B(t_j, (5+2\sqrt{3})r)$. Define S_1^* as the centers s of S^* such that $B(s,r) \subseteq U_{|S_1|}$. To argue $|S_1^*| \geq |S_1|$, it suffices to argue that for all i there exists some $s \in S^*$ such that $B(s,r) \not\subseteq U_{i-1}$ while $B(s,r) \subseteq U_i$ (i.e. s gets charged uniquely to t_i). Now there must be some center $s \in S^*$ such that $B(s, r_{opt}) \cap C_i \neq \emptyset$, as S^* covers $\mathcal{C}$ with radius r_{opt}. Moreover, since $r \geq r_{opt}$, we have $B(s,r) \not\subseteq U_{i-1}$, since otherwise it implies $U_{i-1} \cap C_i \neq \emptyset$ and thus t_i could not have been selected in the ith round as the algorithm had already removed it from P. Conversely, $B(s,r) \subseteq U_i$, since $B(s,r)$ intersects C_i and C_i has radius $\leq (3+2\sqrt{3})r$, and thus $B(s,r) \subseteq B(t_i, (5+2\sqrt{3})r) \subseteq U_i$. Therefore $|S_1^*| \geq |S_1|$.

For any $s \in S_1^*$, $B(s,r) \subseteq U_{|S_1|}$, and since the disks of $\mathcal{C}'$ do not intersect $U_{|S_1|}$, in the optimal solution $\mathcal{C}'$ must be r_{opt}-covered only using centers from $S^* \setminus S_1^*$. Let S_2^* be the subset of centers from $S^* \setminus S_1^*$ which r_{opt} covers $\mathcal{C}' \setminus \mathcal{C}''$. Since any disk $C \in (\mathcal{C}' \setminus \mathcal{C}'')$ has distance $> 2r$ to its nearest neighbor in $\mathcal{C}' \setminus \{C\}$ and $r_{opt} \leq r$, the optimal solution must use a distinct center to cover each disk in $\mathcal{C}' \setminus \mathcal{C}''$, i.e. $|S_2^*| \geq |S_2|$, and moreover, $\mathcal{C}''$ must be covered in the optimal solution by $S^* \setminus (S_1^* \cup S_2^*)$. So finally, let S_3^* be the subset of centers from $S^* \setminus (S_1^* \cup S_2^*)$ which r_{opt} covers $\mathcal{C}''$. By construction, the radius of each $C \in \mathcal{C}''$ is $\geq (3+2\sqrt{3})r$. Thus, by Lemma 7 any point from S_3^* can $(2/\sqrt{3}-1) \cdot (3+2\sqrt{3})r = r \geq r_{opt}$ cover at most 2 disks from $\mathcal{C}''$. Now the graph G, for which our algorithm computes a minimum edge cover $\mathcal{E}$, contains an edge for every pair of disks which can be simultaneously covered with a single r radius ball. Therefore $|S_3^*| \geq |\mathcal{E}| = |S_3|$.

For the running time, computing the set P takes $O(n)$ time. Selecting a new point $p \in P$ and removing all disks from $\mathcal{C}$ which intersect $B(p, (5+2\sqrt{3})r)$ can be done in $O(n)$ time, and thus repeating this till P is empty takes $O(n^2)$ time. Determining the subset $\mathcal{C}''$, and hence the graph G, can naively be done in $O(n^2)$ by checking the distances between all pairs in $\mathcal{C}'$. Selecting a point from each $C \in (\mathcal{C}' \setminus \mathcal{C}'')$ takes $O(n)$ time. Finally, since computing a minimum edge cover can be reduced to computing a maximum matching, $\mathcal{E}$ can be found in $O(n^{2.5})$ time (see [28]). ◀

We remark that it should be possible to improve the running time of the above decision procedure, by arguing that the graph G it constructs is sparse. However, ultimately that will not improve the running time of the following optimization procedure, as it searches over the $O(n^3)$ sized set of Lemma 2.

▶ **Theorem 9.** *There is an $O(n^3 \log n)$ time $(5+2\sqrt{3})$-radius-approximation algorithm for Problem 1, when restricted to instances where $\mathcal{C}$ is a set of disjoint disks.*

Proof. By Lemma 2, in $O(n^3 \log n)$ time we can compute an $O(n^3)$ sized set R of values, such that $r_{opt} \in R$, where r_{opt} is the optimal radius. So sort the values in R, and then binary search over them using the $(5+2\sqrt{3})$-decider of Lemma 8, which we denote $\mathsf{decider}(r)$. Specifically, if $\mathsf{decider}$ returns False we recurse to the right, and if it returns a solution (i.e. True) then we recurse on the left. Note that since our decision procedure is approximate, the values for which it returns True or for which it returns False may not be contiguous in the sorted order of R. Regardless, however, our binary search allows us to find a pair $r' < r$ which are consecutive in R and such that $\mathsf{decider}(r')$ is False, and $\mathsf{decider}(r)$ is True. (Unless $\mathsf{decider}$ always returns True, in which case it returns the smallest value in R.) By Lemma 8

decider is a $(5+2\sqrt{3})$-decider, and thus since decider(r') is False by definition we have that $r' < r_{opt}$. However, as $r' < r$ are consecutive in the sorted order of R and since $r_{opt} \in R$, this implies $r_{opt} \geq r$. On the other hand, again by the definition of a $(5+2\sqrt{3})$-decider, decider(r) outputs a solution with radius at most $(5+2\sqrt{3})r \leq (5+2\sqrt{3})r_{opt}$, thus giving us a $(5+2\sqrt{3})$-approximation as claimed.

By Lemma 2, computing and sorting the $O(n^3)$ values in R takes $O(n^3 \log n)$ time. By Lemma 8 each call to decider takes $O(n^{2.5})$ time, and since we are binary searching over $O(n^3)$ values, the time for all calls to decider is $O(n^{2.5}\log(n^3)) = O(n^{2.5}\log n)$. Thus the total time is $O(n^3 \log n)$ as claimed. ◀

Our focus in this paper is on the planar case, however, in the full version [22] we remark how the above decision procedure works in higher dimensions. The above optimization procedure does not immediately extend as it makes use of Lemma 2, however, in the full version we informally sketch how one can approximately recover the same result.

6 An Efficient FPTAS for Bounded k

By Lemma 2, we can compute a set of $O(n^3)$ points which contains a subset of size k that is an optimal k-center solution. Thus, for constant k, enumerating all $O(n^{3k})$ possible subsets, and taking the minimum cost found, yields a polynomial time algorithm. In this section, we argue that for constant k, we can achieve a $(1+\varepsilon)$-radius-approximation for unit disks, whose running time depends only linearly on n. Contrast this with Theorem 6, where we argued that when k is not assumed to be constant, that the problem is hard to approximate for unit disks within a given constant factor. We use the following from Agarwal and Procopiuc [3].

▶ **Theorem 10** ([3]). *Given a set P of n points in the plane, there is an $O(n\log k)+(k/\varepsilon)^{O(\sqrt{k})}$ time $(1+\varepsilon)$-radius-approximation algorithm for k-center, denoted* **kCenter**(ε, P).

▶ **Theorem 11.** *There is an $O(n\log k)+(k/\varepsilon)^{O(k)}$ time $(1+\varepsilon)$-radius-approximation algorithm for Problem 1, when restricted to instances where $\mathcal{C}$ is a set of disjoint unit disks.*

Proof. Let P denote the set of center points of the disks in $\mathcal{C}$. For any given set S of k points in the plane, let $r_P(S) = \max_{p\in P} ||p-S||$ and $r_{\mathcal{C}}(S) = \max_{C\in\mathcal{C}} ||C-S||$. Observe that $r_{\mathcal{C}}(S) \leq r_P(S) \leq r_{\mathcal{C}}(S)+1$. Specifically, $r_{\mathcal{C}}(S) \leq r_P(S)$ since any ball (in particular one centered at a point from S) which contains a center point from P also intersects the corresponding disk in $\mathcal{C}$. On the other hand, $r_P(S) \leq r_{\mathcal{C}}(S)+1$ since for any ball intersecting a disk in $\mathcal{C}$, if we increase its radius by 1 then it will contain the center point of that disk, as $\mathcal{C}$ consists of unit disks.

Let r_{opt} denote the optimum radius for the given instance $\mathcal{C}, k$ of Problem 1. We consider two cases based on the value of r_{opt}. First, suppose that $r_{opt} > 2/\varepsilon$. Let S' denote the solution returned by **kCenter**$(\varepsilon/3, P)$. By the above inequalities and Theorem 10,

$$r_{\mathcal{C}}(S') \leq r_P(S') \leq (1+\varepsilon/3)\min_{S\subset\mathbb{R}^2,|S|=k} r_P(S) \leq (1+\varepsilon/3)(1+\min_{S\subset\mathbb{R}^2,|S|=k} r_{\mathcal{C}}(S))$$
$$= (1+\varepsilon/3)(1+r_{opt}) < (1+\varepsilon/3)(\varepsilon r_{opt}/2 + r_{opt}) = (1+\varepsilon/3)(1+\varepsilon/2)r_{opt} \leq (1+\varepsilon)r_{opt},$$

where the last inequality assumed $\varepsilon \leq 1$. Thus S' is $(1+\varepsilon)$-approximation for Problem 1.

Now suppose that $r_{opt} \leq 2/\varepsilon$. In this case observe that for any point $x \in \mathbb{R}^2$, the ball $B(x, r_{opt})$ can intersect only $O(1/\varepsilon^2)$ disks from $\mathcal{C}$ as they are disjoint and all have radius 1. Thus any center from the optimal solution can cover at most $O(1/\varepsilon^2)$ disks within the optimal radius, and so it must be that $n = O(k/\varepsilon^2)$.

The algorithm is now straightforward. If $n \leq \gamma k/\varepsilon^2$, for some sufficiently large constant γ, then by Lemma 2 in $O((k/\varepsilon^2)^3 \log(k/\varepsilon))$ time we can compute a set P of $O((k/\varepsilon^2)^3)$ points such that P contains an optimal set of k centers. We try all possible subsets of P of size k and take the best one. There are $O((k/\varepsilon^2)^{3k})$ such subsets, and for each subset its cost can be determined in $O(kn) = O((k/\varepsilon)^2)$ time. Thus in this case we can compute the optimal solution in $O((k/\varepsilon)^2 \cdot (k/\varepsilon^2)^{3k}) = (k/\varepsilon)^{O(k)}$ time.

On the other hand, if $n > \gamma k/\varepsilon^2$ then the above implies $r_{opt} > 2/\varepsilon$. In this case it was argued above that **kCenter**$(\varepsilon/3, P)$ returns a $(1+\varepsilon)$-approximation, and by Theorem 10 it does so in $O(n \log k) + (k/\varepsilon)^{O(\sqrt{k})}$ time. In either case, we have a $(1+\varepsilon)$-approximation (or better) and the total time is $\max\{(k/\varepsilon)^{O(k)}, O(n \log k) + (k/\varepsilon)^{O(\sqrt{k})}\}$. ◀

7 One Dimensional Clustering with Neighborhoods

In this section we show that despite clustering with neighborhoods being hard to radius approximate within any factor in the plane, we can solve the one dimensional variant exactly in $O(n \log n)$ time, even when object intersections are allowed. First, we argue the decision problem can be solved in linear time. Then we argue that we can use a scheme similar to that in [15] to search for the optimal radius.

In one dimension, a convex object is just a closed interval. Thus we have the following one dimensional version of Problem 1, where intersections are no longer prohibited.

▶ **Problem 12** (One Dimensional Clustering with Neighborhoods). *Given a set $\mathcal{C}$ of n closed intervals on the real line, and an integer parameter $k \geq 0$, find a set of k points S (called centers) which minimize the maximum distance to an interval in $\mathcal{C}$. That is,*

$$S = \arg \min_{S' \subset \mathbb{R}, |S'|=k} \max_{C \in \mathcal{C}} ||C - S'||.$$

The following decision procedure is similar in spirit to various folklore results for interval problems in one dimension (for example, see the discussion in [13] on interval stabbing). The challenge is turning this decision procedure into an efficient optimization procedure, for which as discussed below we make use of [15].

We first sort the intervals in increasing order both by their left and by their right endpoints. We maintain cross links between the two sorted lists so that if we remove an interval from one list, its copy in the other list can be removed in constant time.

▶ **Lemma 13.** *Given an instance $\mathcal{C}, k$ of Problem 12, where the intervals have been presorted, for any query radius r, in $O(n)$ time one can decide whether $r \geq r_{opt}$.*

Proof. We build a set S of centers as follows, where initially $S = \emptyset$. Let $[\alpha, \beta]$ denote the interval with the leftmost right endpoint (i.e. β is smallest among all intervals). We place a center at $\beta + r$ and add it S. Next we remove all intervals which intersect the ball $B(\beta + r, r)$. Note that these intersecting intervals are precisely those whose left endpoint is $\leq \beta + 2r$, as this condition is clearly necessary to intersect $B(\beta + r, r)$, but also sufficient as all intervals have right end point $\geq \beta$. We then repeat this process until all intervals are removed. If $|S| \leq k$ we return True and otherwise we return False.

Observe that every time we place a center, we remove intervals it covers within distance r. Thus the final set S is a set of centers of radius r, and so if $|S| \leq k$, then $r \geq r_{opt}$ and the algorithm correctly returns True. Moreover, we now argue that S is a minimum cardinality set of centers of radius r, and thus if $|S| > k$ then the algorithm correctly returns False. Adopting notation from above, let $[\alpha, \beta]$ be the interval with leftmost right endpoint, and

let c be the center our algorithm places at $\beta + r$. Now in the minimum cardinality solution, there must be at least one center c' within distance r from $[\alpha, \beta]$, implying the location of c' is $\leq \beta + r$. Thus c' can only r-cover intervals with left endpoint $\leq \beta + 2r$. However, as described above, c r-covers all intervals with left endpoint $\leq \beta + 2r$, and thus c' r-covers a subset of those c does. Conversely, the subset of intervals not r-covered by c is a subset of those not r-covered by c'. By induction our algorithm uses the smallest possible number of centers to r-cover the intervals not r-covered by c, which therefore is at most the number centers the global minimum solution uses to r-cover the superset of intervals not r-covered by c'. Thus overall our set of centers was an r-cover of minimum cardinality.

For the running time, observe that determining the location of the next center takes constant time since it only depends on the leftmost right endpoint, and we assumed we have the sorted ordering of the intervals by right endpoint. Moreover, we can remove all of the intervals intersecting the r radius ball at the new center in time linear in the number of intersecting intervals, since as discussed above these intersecting intervals are a prefix of the sorted ordering by left endpoint. As we spend constant time per interval removed, overall this is an $O(n)$ time algorithm. ◀

Lemma 13 gives us a decision procedure for Problem 12 which we now wish to utilize to search for the optimum radius. We use the following lemma to reduce the search space, which can be seen as a simplification of Lemma 2 for the one dimensional case, where here we only need to consider distances from bisecting points rather than bisecting curves.

▶ **Lemma 14.** *Let $\mathcal{C}$ be a set of closed intervals. Then for any value k, the optimal radius for the instance $\mathcal{C}, k$ of Problem 12 is either 0 or $||C - C'||/2$ for some pair $C, C' \in \mathcal{C}$.*

Proof. For any value k, let S be an optimal solution with optimal radius r_{opt}. Consider an arbitrary center $s \in S$, and let $\mathcal{C}'$ be the subset of $\mathcal{C}$ which intersects the ball $B(s, r_{opt})$. We can assume that $|\mathcal{C}'| \geq 1$, as otherwise $B(s, r_{opt})$ does not intersect any interval and so s can be thrown out. If $|\mathcal{C}'| = 1$, then s intersects only one interval, and thus without loss of generality s can be placed inside this interval, i.e. at distance 0 from it. So assume $|\mathcal{C}'| > 1$, and let C be the furthest interval from s in $\mathcal{C}'$. As we move s towards C, so long as C remains the furthest interval from s in $\mathcal{C}'$, $B(s, ||s - C||)$ will continue to intersect all intervals in $\mathcal{C}'$. If C always remains the furthest, when s eventually reaches C, its distance to C and hence all of $\mathcal{C}'$ will be 0. Otherwise, if before we reach C, s is no longer the furthest from s, then we must have crossed the bisector point between C and some other interval in $\mathcal{C}'$. In this case, we can place s on this bisector point and $B(s, ||s - C||)$ will intersect all intervals in $\mathcal{C}'$, and moreover $||s - C|| \leq r_{opt}$ since $||s - C||$ monotonically decreased as we moved s towards C. Modifying all centers in S in this way thus produces a solution whose radius is $\leq r_{opt}$ and is either 0 or the distance from a bisector point to either interval in the pair it bisects. ◀

Given a set $\mathcal{C}$ of n intervals, let $P(\mathcal{C})$ denote the set of all $2n$ left and right endpoints of the intervals in $\mathcal{C}$. To find the optimal solution to an instance $\mathcal{C}, k$ of Problem 12, by Lemma 14, we can binary search over the interpoint distances of points in $P(\mathcal{C})$ using our decider from Lemma 13. (When we call the decider we divide the interpoint distance by two as Lemma 14 actually tells us it is a bisector distance.) As there are $\Theta(n^2)$ interpoint distances, naively this approach takes $O(n^2 \log n)$ time. However, [15] previously showed that in the abstract setting where one is given a linear time decider, and the optimal solution is an interpoint distance, one can find the optimal solution in $O(n \log n)$ time. This is achieved by reducing the problem to searching in an implicitly defined sorted matrix, which we describe in the full version [22]. Below is the summarizing theorem.

▶ **Theorem 15.** *Problem 12 can be solved in $O(n \log n)$ time, where $n = |\mathcal{C}|$.*

References

1. P. K. Agarwal, J. Matousek, and M. Sharir. On range searching with semialgebraic sets II. In *53rd Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 420–429, 2012. doi:10.1109/FOCS.2012.32.
2. P. K. Agarwal, J. Pach, and M. Sharir. State of the union (of geometric objects). In J.E. Goodman, J. Pach, and R. Pollack, editors, *Surveys on Discrete and Computational Geometry: Twenty Years Later*, volume 453 of *Contemp. Math.*, pages 9–48. AMS, 2008.
3. P. K. Agarwal and C. M. Procopiuc. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2):201–226, 2002. doi:10.1007/s00453-001-0110-y.
4. H.-K. Ahn, S.-S. Kim, C. Knauer, L. Schlipf, C.-S. Shin, and A. Vigneron. Covering and piercing disks with two centers. *Comput. Geom.*, 46(3):253–262, 2013.
5. G. Anegg, H. Angelidakis, A. Kurpisz, and R. Zenklusen. A technique for obtaining true approximations for k-center with covering constraints. In *21st Integer Programming and Combinatorial Optimization (IPCO)*, volume 12125 of *LNCS*, pages 52–65. Springer, 2020. doi:10.1007/978-3-030-45771-6_5.
6. M. Badoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. In *34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 250–257. ACM, 2002. doi:10.1145/509907.509947.
7. S. Bandyapadhyay, T. Inamdar, S. Pai, and K. R. Varadarajan. A constant approximation for colorful k-center. In *27th Annual European Symposium on Algorithms (ESA)*, volume 144 of *LIPIcs*, pages 12:1–12:14, 2019. doi:10.4230/LIPIcs.ESA.2019.12.
8. M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *12th Annual Symposium on Discrete Algorithms (SODA)*, pages 642–651. ACM/SIAM, 2001. URL: http://dl.acm.org/citation.cfm?id=365411.365555.
9. O. Cheong, H. Everett, M. Glisse, J. Gudmundsson, S. Hornus, S. Lazard, M. Lee, and H.-S. Na. Farthest-polygon Voronoi diagrams. *Comput. Geom.*, 44(4):234–247, 2011.
10. A. Dumitrescu and J. S. B. Mitchell. Approximation algorithms for TSP with neighborhoods in the plane. *J. Algorithms*, 48(1):135–159, 2003. doi:10.1016/S0196-6774(03)00047-6.
11. D. Eisenstat, P. N. Klein, and C. Mathieu. Approximating k-center in planar graphs. In *25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 617–627. SIAM, 2014. doi:10.1137/1.9781611973402.47.
12. T. Feder and D. H. Greene. Optimal algorithms for approximate clustering. In *20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 434–444. ACM, 1988. doi:10.1145/62212.62255.
13. S. P. Fekete, K. Huang, J. S. B. Mitchell, O. Parekh, and C. A. Phillips. Geometric hitting set for segments of few orientations. *Theory Comput. Syst.*, 62(2):268–303, 2018.
14. H. De Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Comb.*, 10(1):41–51, 1990. doi:10.1007/BF02122694.
15. G. N. Frederickson. Parametric search and locating supply centers in trees. In *2nd Workshop on Algorithms and Data Structures (WADS)*, pages 299–319, 1991. doi:10.1007/BFb0028271.
16. J. Gao, M. Langberg, and L. J. Schulman. Analysis of incomplete data and an intrinsic-dimension helly theorem. *Discrete and Computational Geometry*, 40(4):537–560, 2008. doi:10.1007/s00454-008-9107-5.
17. J. Gao, M. Langberg, and L. J. Schulman. Clustering lines in high-dimensional space: Classification of incomplete data. *ACM Trans. Algorithms*, 7(1):8:1–8:26, 2010. doi:10.1145/1868237.1868246.
18. M. R. Garey and D. S. Johnson. The rectilinear steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977. doi:10.1137/0132071.
19. T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.
20. D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985. doi:10.1287/moor.10.2.180.

21 W. Hsu and G. L. Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979. doi:10.1016/0166-218X(79)90044-1.

22 H. Huang, G. Klimenko, and B. Raichel. Clustering with neighborhoods, September 2021. arXiv:2109.13302.

23 X. Jia, K. Sheth, and O. Svensson. Fair colorful k-center clustering. In *21st Integer Programming and Combinatorial Optimization (IPCO)*, volume 12125 of *LNCS*, pages 209–222. Springer, 2020. doi:10.1007/978-3-030-45771-6_17.

24 M. I. Karavelas and M. Yvinec. The Voronoi diagram of planar convex objects. In *11th Annual European Symposium on Algorithms (ESA)*, volume 2832 of *LNCS*, pages 337–348. Springer, 2003. doi:10.1007/978-3-540-39658-1_32.

25 R. Klein. *Concrete and Abstract Voronoi Diagrams*, volume 400 of *LNCS*. Springer, 1989. doi:10.1007/3-540-52055-4.

26 E. Lee and L. J. Schulman. Clustering affine subspaces: Hardness and algorithms. In *24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 810–827. SIAM, 2013. doi:10.1137/1.9781611973105.58.

27 Y. Marom and D. Feldman. k-means clustering of lines for big data. In *32nd Annual Advances in Neural Information Processing Systems (NeurIPS)*, pages 12797–12806, 2019. URL: http://papers.nips.cc/paper/9442-k-means-clustering-of-lines-for-big-data.

28 S. Micali and V. V. Vazirani. An $O(\sqrt{|V|}|E|)$ algoithm for finding maximum matching in general graphs. In *21st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 17–27, 1980.

29 N. H. Mustafa and S. Ray. Improved results on geometric hitting set problems. *Discrete and Computational Geometry*, 44(4):883–895, 2010. doi:10.1007/s00454-010-9285-9.

30 R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *29th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 475–484. ACM, 1997. doi:10.1145/258533.258641.

31 G. Xu and J. Xu. Efficient approximation algorithms for clustering point-sets. *Computational Geometry*, 43(1):59–66, 2010. doi:10.1016/j.comgeo.2007.12.002.