

# Clustering with Few Disks to Minimize the Sum of Radii

Mikkel Abrahamsen ✉

University of Copenhagen, Denmark

Sarita de Berg ✉

Utrecht University, The Netherlands

Lucas Meijer ✉

Utrecht University, The Netherlands

André Nusser ✉

University of Copenhagen, Denmark

Leonidas Theocharous ✉

Eindhoven University of Technology, The Netherlands

---

## Abstract

Given a set of  $n$  points in the Euclidean plane, the  $k$ -MINSUMRADIUS problem asks to cover this point set using  $k$  disks with the objective of minimizing the sum of the radii of the disks. After a long line of research on related problems, it was finally discovered that this problem admits a polynomial time algorithm [GKKPV '12]; however, the running time of this algorithm is  $\mathcal{O}(n^{881})$ , and its relevance is thereby mostly of theoretical nature. A practically and structurally interesting special case of the  $k$ -MINSUMRADIUS problem is that of small  $k$ . For the 2-MINSUMRADIUS problem, a near-quadratic time algorithm with expected running time  $\mathcal{O}(n^2 \log^2 n \log^2 \log n)$  was given over 30 years ago [Eppstein '92].

We present the first improvement of this result, namely, a near-linear time algorithm to compute the 2-MINSUMRADIUS that runs in expected  $\mathcal{O}(n \log^2 n \log^2 \log n)$  time. We generalize this result to any constant dimension  $d$ , for which we give an  $\mathcal{O}(n^{2-1/(\lceil d/2 \rceil + 1) + \epsilon})$  time algorithm. Additionally, we give a near-quadratic time algorithm for 3-MINSUMRADIUS in the plane that runs in expected  $\mathcal{O}(n^2 \log^2 n \log^2 \log n)$  time. All of these algorithms rely on insights that uncover a surprisingly simple structure of optimal solutions: we can specify a linear number of lines out of which one separates one of the clusters from the remaining clusters in an optimal solution.

**2012 ACM Subject Classification** Theory of computation → Computational geometry

**Keywords and phrases** geometric clustering, minimize sum of radii, covering points with disks

**Digital Object Identifier** 10.4230/LIPIcs.SoCG.2024.2

**Related Version** *Full Version:* <https://arxiv.org/abs/2312.08803>

**Funding** *Mikkel Abrahamsen:* is supported by Starting Grant 1054-00032B from the Independent Research Fund Denmark under the Sapere Aude research career programme and is part of Basic Algorithms Research Copenhagen (BARC), supported by the VILLUM Foundation grant 16582.

*Lucas Meijer:* is supported by the Netherlands Organisation for Scientific Research (NWO) under project no. VI.Vidi.213.150.

*André Nusser:* is part of Basic Algorithms Research Copenhagen (BARC), supported by the VILLUM Foundation grant 16582.

*Leonidas Theocharous:* is supported by the Dutch Research Council (NWO) through Gravitation-grant NETWORKS-024.002.003.

**Acknowledgements** This work was initiated at the Workshop on New Directions in Geometric Algorithms, May 14–19 2023, Utrecht, The Netherlands.



© Mikkel Abrahamsen, Sarita de Berg, Lucas Meijer, André Nusser, and Leonidas Theocharous;

licensed under Creative Commons License CC-BY 4.0

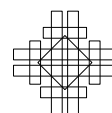
40th International Symposium on Computational Geometry (SoCG 2024).

Editors: Wolfgang Mulzer and Jeff M. Phillips; Article No. 2; pp. 2:1–2:15



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



## 1 Introduction

Clustering seeks to partition a data set in order to obtain a deeper understanding of its structure. There are different clustering notions that cater to different applications. An important subclass is geometric clusterings [9]. In their general form, as defined in [9], geometric clusterings try to partition a set of input points in the plane into  $k$  clusters such that some objective function is minimized. More formally, let  $f$  be a symmetric  $k$ -ary function and  $w$  a non-negative function over all subsets of input points. The geometric clustering problem is defined as follows: Given a set  $P$  of points in the Euclidean plane and an integer  $k$ , partition  $P$  into  $k$  sets  $C_1, \dots, C_k$  such that  $f(w(C_1), \dots, w(C_k))$  is minimized. Popular choices for the weight function  $w$  are the radius of the minimum enclosing disk and the sum of squared distances from the points to the mean. The function  $f$  aggregates the weights of all clusters, for example using the maximum or the sum.

Arguably the most popular types of clustering in this setting are  $k$ -CENTER clustering (with  $f$  being the maximum, and  $w$  being the radius of the minimum enclosing disk) and  $k$ -MEANS clustering (with  $f$  being the sum, and  $w$  being the sum of squared distances to the mean). While geometric clusterings have the advantage that their underlying objective function can be very intuitive, unfortunately the cluster boundaries might sometimes be slightly more complex. For example, the disks defining the clusters can have a large overlap in  $k$ -CENTER clustering, while in  $k$ -MEANS clustering the boundaries are defined by the Voronoi diagram on the mean points of the clusters (whose complexity has an exponential dependency on the dimension). An instance of geometric clustering for which the cluster boundaries implicitly consist of non-overlapping disks is  $k$ -MINSUMRADIUS. In  $k$ -MINSUMRADIUS clustering we want to minimize the sum of radii of the  $k$  disks with which we cover the input point set. In the geometric clustering setting, this means that the function  $f$  is the sum and  $w$  is the radius of the minimum enclosing disk. This is the notion of clustering that we consider in this work.

While  $k$ -CENTER and  $k$ -MEANS are both NP-hard in the Euclidean plane when  $k$  is part of the input [24, 26], the  $k$ -MINSUMRADIUS problem can be solved in polynomial time [18]. Interestingly, this result extends to any Euclidean space when its dimension is considered a constant. However, the bound on the running time<sup>1</sup> is  $\mathcal{O}(n^{881})$ . Whether there exists a practically fast exact algorithm for  $k$ -MINSUMRADIUS can be considered one of the most interesting questions in geometric clustering. While the running time of the known polynomial-time algorithm can likely be slightly improved using the same techniques, the balanced separators that are used in the algorithm inevitably lead to a high running time. Thus, we believe that further structural insights into the problem – especially with respect to separators – are needed to greatly reduce the exponent of the polynomial running time.

In order to obtain a deeper understanding of the problem structure and due to its practical relevance, in this work we consider the  $k$ -MINSUMRADIUS problem for small values of  $k$ . The importance of this setting is reflected in the extensive work that was conducted in the analogous setting for the  $k$ -CENTER and  $k$ -MEANS problem, especially for the case of two clusters [4, 5, 10, 13, 14, 17, 22, 29, 31] – also called *bi-partition*. Bi-partition problems are

---

<sup>1</sup> As the radii of minimum enclosing disks can contain square roots, the value of a solution is a sum of square roots. However, it is not known how to compare two sums of square roots in polynomial time in the number of elements. The running time merely counts the number of such comparisons. Additionally, the setting is slightly different than ours in that it assumes that the centers of the clusters have to be located at input points. We believe that this algorithm also carries over to the setting of unrestricted cluster centers, incurring some additional polynomial factors in the running time.



■ **Figure 1** The optimal 2-CENTER clustering (left) compared to the optimal 2-MINSUMRADIUS clustering (right) for the same point set. In this example 2-MINSUMRADIUS clustering better captures the structure of the point set than 2-CENTER clustering.

of interest on their own, but they can additionally be used as a subroutine in hierarchical clustering methods. Even more interestingly, while near-linear time algorithms for 2-CENTER clustering received a lot of attention [10, 17, 13, 29, 31], the best known algorithm for 2-MINSUMRADIUS still has near-quadratic expected running time  $\mathcal{O}(n^2 \log^2 n \log^2 \log n)$  [16], which has seen no improvement in the last 30 years, despite significant work on related problems.

We are aware of no reason why 2-CENTER should deserve more attention than 2-MINSUMRADIUS, and we believe that the preference for the former problem may have arisen merely out of habit or historical trends in research focus. The idea that similar data points are grouped together in the same cluster lies at the heart of clustering and, in particular, this is the reason why clustering is used for classification tasks in machine learning. We note that in this regard, as shown in Figure 1, optimal solutions to 2-MINSUMRADIUS often result in more natural clusters than for 2-CENTER.

## Our results

In this work, we break the quadratic barrier for 2-MINSUMRADIUS in the Euclidean plane by presenting a near-linear time algorithm with expected running time  $\mathcal{O}(n \log^2 n \log^2 \log n)$ . Our method actually extends to any constant dimension  $d \geq 3$ , again yielding a subquadratic algorithm with running time  $\mathcal{O}(n^{2-1/(\lceil d/2 \rceil + 1) + \epsilon})$ . Moreover, we extend our structural insights to planar 3-MINSUMRADIUS – matching the previously best 2-MINSUMRADIUS running time – and give an algorithm with expected  $\mathcal{O}(n^2 \log^2 n \log^2 \log n)$  running time, which is the first non-trivial result on this special case that we are aware of. The running time for planar 2-MINSUMRADIUS and 3-MINSUMRADIUS can be made deterministic by using the deterministic algorithm to maintain a minimum enclosing ball [16], which increases the running times to  $\mathcal{O}(n \log^4 n)$  and  $\mathcal{O}(n^2 \log^4 n)$ , respectively.

The main technical contribution leading to these results are structural insights that simplify the problem significantly. Concretely, we show that the points on the boundary of the minimum enclosing disk (or ball, in higher dimensions) of the point set induce a constant number of directions such that there is a line (or hyperplane) with one of those directions that separates one cluster from the other clusters in an optimal solution. As there are only linearly many combinatorially distinct separator lines for each direction, we have linearly many separators in total that we have to consider. Note that this is the main difference to the previously best algorithm for planar 2-MINSUMRADIUS [16], which considered quadratically many separators. We then check all clusterings induced by these separators using an algorithm from [16] to dynamically maintain a minimum enclosing disk and, in the  $k = 3$  case, we use our 2-MINSUMRADIUS algorithm as subroutine. For the higher-dimensional 2-MINSUMRADIUS problem, we similarly use an algorithm to maintain a

minimum enclosing ball in any dimension  $d \geq 3$  [3]. While our algorithms are interesting in their own right, we additionally hope to enable a better understanding of the general case by uncovering this surprisingly simple structure of separators in the cases  $k \in \{2, 3\}$ .

In Section 3, we discuss the near-linear time algorithm for 2-MINSUMRADIUS, including the sub-quadratic time algorithm for  $d \geq 3$ . And in Section 4 we discuss the near-quadratic time algorithm for 3-MINSUMRADIUS. We conclude our work in Section 5.

### Other related work

There is a multitude of work on more general or similar problems. In a general metric space, constant-factor approximation algorithms for  $k$ -MINSUMRADIUS have been known for a long time [12]. Recently, a 3-approximation algorithm was presented, which also works in the setting where outliers are allowed [8].

In [9], the authors consider the problem of clustering a set of points into  $k$  clusters with an objective function that is monotonically increasing in the diameters or radii of the clusters. Concretely, they give an algorithm with running time  $\mathcal{O}(n^{6k})$ . In the same work, the authors consider separators of these types of clusterings. However, as opposed to our work, the authors rather focus on the existence of separators in their general setting, while we try to reduce the number of relevant separators to consider in  $k$ -MINSUMRADIUS.

In [6], a generalized version of  $k$ -MINSUMRADIUS is studied, namely, that of minimizing the sum of radii taken to the power of  $\alpha$ . The authors show NP-hardness for  $\alpha > 1$  in this setting, but they give a polynomial-time algorithm for the special case in which all centers of disks have to lie on a line.

Another similar setting is that of minimizing the sum of diameters of the clusters, which we call  $k$ -MINSUMDIAMETER. We want to emphasize that when referring to the radius of a cluster, this refers to the radius of the minimum enclosing disk, while when referring to the diameter, it commonly refers to the distance between the furthest pair of points in the cluster. In contrast to the case of  $k$ -MINSUMRADIUS, in  $k$ -MINSUMDIAMETER the cluster boundaries do not behave as nicely. For the 2-MINSUMDIAMETER problem, a near-linear time algorithm in the Euclidean plane was given more than 30 years ago [20]. However, for this problem, different techniques seem to be relevant, as farthest-point Voronoi diagrams are utilized rather than separators. A similar problem to  $k$ -MINSUMDIAMETER is the problem of minimizing the maximum diameter of the clusters, which has a near-quadratic time algorithm for the case of three clusters [19].

Opposed to enclosing points using disks, there is a class of problems that aims at enclosing a set of points with multiple closed polylines while minimizing the sum of the perimeters. This problem and similar variants were studied in the general setting of arbitrarily many clusters [1] as well as in the setting of two clusters [2].

Finally, very recently there was also some work to determine the fine-grained running times of the  $k$ -CENTER problem – partly for other shapes than disks – for small values of  $k$  [11].

## 2 Preliminaries

For a set of points  $Q \subset \mathbb{R}^d$ , let  $\text{MEB}(Q)$  be the minimum enclosing ball that contains all points of  $Q$ . We denote by  $r(Q)$  the radius of  $\text{MEB}(Q)$ . For  $d = 2$ , we denote the minimum enclosing ball, which is a disk, by  $\text{MED}(Q)$ .

► **Definition 1** (*k*-MINSUMRADIUS). Let  $P$  be a set of  $n$  points in  $\mathbb{R}^d$  and  $k$  be a positive integer. The *k*-MINSUMRADIUS problem asks to partition  $P$  into  $k$  clusters  $C_1, C_2, \dots, C_k$  such that  $\sum_{i=1}^k r(C_i)$  is minimized.

Throughout the paper, we let  $D$  denote the minimum enclosing ball of the input set of our *k*-MINSUMRADIUS instance, i.e.,  $D := \text{MEB}(P)$ , and let  $c$  be the center of  $D$ . We say that a point set  $Q$  defines a ball  $D'$  if  $\text{MEB}(Q) = D'$ . Given  $D$  and a point  $p$  on the boundary of this ball, we define  $p^*$  to be the diametrically-opposing point of  $p$  on  $D$ . We define  $d(p)$  to be the diameter of  $D$  that contains  $p$ , i.e.,  $d(p)$  is the segment from  $p$  to  $p^*$ .

The next lemma states that in an optimal solution clusters are well-separated, in the sense that their minimum enclosing balls are disjoint.

► **Lemma 2.** Given a *k*-MINSUMRADIUS instance, there exists an optimal solution with clusters  $C_1, \dots, C_k$  such that  $\text{MEB}(C_i) \cap \text{MEB}(C_j) = \emptyset$  for all distinct  $i, j \in [k]$ .

**Proof.** Consider an arbitrary optimal solution and assume that there are two clusters  $C_i, C_j$  for which  $\text{MEB}(C_i) \cap \text{MEB}(C_j) \neq \emptyset$ . We can replace the clusters  $C_i, C_j$  by a single cluster  $C' := C_i \cup C_j$ . Note that this does not increase the total clustering cost as  $r(C') \leq r(C_i) + r(C_j)$ . Hence, we obtain a clustering with one less cluster (or, in other words, one more empty cluster). We can recursively apply this argument until we either end up with a single cluster (which trivially satisfies the lemma) or all clusters have pairwise non-overlapping minimum enclosing balls. ◀

We also make use of the following (slightly differently phrased) known lemma, which intuitively states that any plane through the center of a minimum enclosing ball has points on both sides.

► **Lemma 3** ([7, Lemma 2.2]). Let  $Q \subset \mathbb{R}^d$  with  $|Q| = d + 1$  be a set of points and let  $\text{MEB}(Q)$  be their minimum enclosing ball. There exists no hyperplane  $H$  through the center of  $\text{MEB}(Q)$  such that all points of  $Q$  are strictly on one side of  $H$ .

We say that a  $(d - 1)$ -dimensional hyperplane  $\ell$  separates two sets  $A, B \subset \mathbb{R}^d$  if  $A$  and  $B$  are contained in different closed half-spaces defined by  $\ell$  and  $\ell \cap A = \emptyset$  or  $\ell \cap B = \emptyset$ . In our algorithms for 2-MINSUMRADIUS and 3-MINSUMRADIUS in the plane, we consider a sweep-line  $\ell$  perpendicular to a direction defined by points on the boundary of  $D$ , keeping track of the points in  $P$  on both sides of  $\ell$ . Our definition of a separator is convenient, as it allows us to include all input points on  $\ell$  together with the points on one of the sides, thus avoiding any general position assumptions.

### 3 Near-linear algorithm for 2-MinSumRadius

In this section, we present our algorithm for 2-MINSUMRADIUS for both the plane and higher dimensions. First, we consider the problem in the plane to build an intuitive understanding, which is also applicable to the 3-MINSUMRADIUS problem, before generalizing to higher dimensions in Section 3.3. For the plane, we prove the following result:

► **Theorem 4.** For a set  $P$  of  $n$  points in the Euclidean plane, an optimal 2-MINSUMRADIUS clustering can be computed in expected  $\mathcal{O}(n \log^2 n \log^2 \log n)$  or worst-case  $\mathcal{O}(n \log^4 n)$  time.

In Section 3.1 we present the algorithm, and in Section 3.2 we prove the main structural result that our algorithm relies on.

■ **Algorithm 1** Near-linear time algorithm for 2-MINSUMRADIUS.

---

```

1: procedure 2-MINSUMRADIUS( $P$ )
2:    $D \leftarrow$  minimum enclosing disk of  $P$ 
3:    $p_1, p_2, p_3 \leftarrow$  points defining  $D$ 
4:    $S_{\text{OPT}} \leftarrow (D, \emptyset)$  ▷ single cluster containing all of  $P$ 
5:   for each diameter  $d(p_i)$  at  $p_i$  do
6:     initialize dynamically maintained MEDs:  $A \leftarrow \text{MED}(\emptyset), B \leftarrow \text{MED}(P)$ 
7:     for each separator orthogonal to  $d(p_i)$  through a  $p \in P$  in sorted order do
8:       update  $A$  adding  $p$ , and update  $B$  removing  $p$ 
9:       if  $(A, B)$  is a better solution than  $S_{\text{OPT}}$  then
10:         $S_{\text{OPT}} \leftarrow (A, B)$ 
11:   return  $S_{\text{OPT}}$ 

```

---

### 3.1 Algorithm

Our algorithm uses the insight that there exist a linear number of separators, such that one of them separates the points in cluster  $C_1$  from those in cluster  $C_2$ .

► **Lemma 5.** *Consider a point set  $P$  in the Euclidean plane and let  $p_1, p_2, p_3$  be three points of  $P$  that define the minimum enclosing disk of  $P$  (with potentially  $p_2 = p_3$ ). There exists an optimal 2-MINSUMRADIUS clustering  $C_1, C_2$  of  $P$  such that for a point  $q \in \{p_1, p_2, p_3\}$  and a line  $\ell$  orthogonal to  $d(q)$ , the line  $\ell$  separates  $C_1$  from  $C_2$ .*

We prove this result in Section 3.2. We now explain our algorithm relying on Lemma 5; see Algorithm 1 for the pseudo-code.

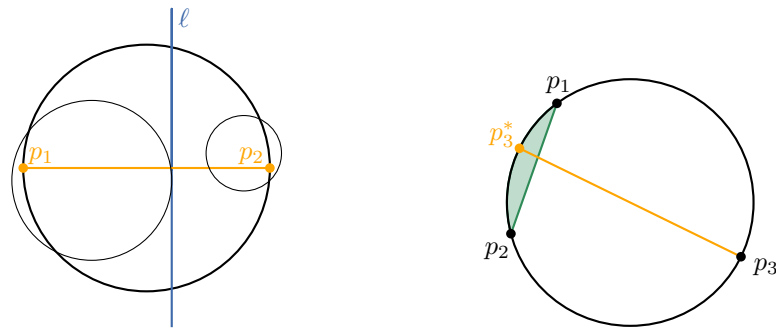
**Algorithm description.** Let  $p_1, p_2, p_3$  denote a triple of points in  $P$  that define  $D$  (with possibly  $p_2 = p_3$ ). These points can be computed in  $\mathcal{O}(n)$  time [25]. We try out every combinatorially distinct line orthogonal to  $d(p_i)$  for  $i \in \{1, 2, 3\}$  as a separator. We consider the separators orthogonal to a specific  $d(p_i)$  in sorted order such that in each step only one point jumps over the separator. This ensures that the minimum enclosing disks on both sides of the separator only change by a single point per step, and we can therefore use a data structure to dynamically maintain them. We then select the best solution found using these separators.

Correctness follows directly from Lemma 5, hence let us consider the running time. We can maintain the minimum enclosing disks  $A$  and  $B$  in expected  $\mathcal{O}(\log^2 n \log^2 \log n)$  or worst-case  $\mathcal{O}(\log^4 n)$  amortized time per update [16]. So, checking all  $n$  separators requires expected  $\mathcal{O}(n \log^2 n \log^2 \log n)$  or  $\mathcal{O}(n \log^4 n)$  worst-case time. As we only have three diameters  $d(p_1), d(p_2), d(p_3)$  to handle, we can compute the optimal solution in the same time.

### 3.2 Linear number of cuts

In this subsection we prove Lemma 5, which states that it suffices to only check separators orthogonal to one of the diameters  $d(p_1), d(p_2)$ , or  $d(p_3)$ .

**Proof of Lemma 5.** Let  $C_1, C_2$  be an optimal clustering. If  $r(C_1) + r(C_2) = r(D)$ , then the clustering  $P, \emptyset$  is also optimal and trivially satisfies the claim. We therefore in the following assume that  $r(C_1) + r(C_2) < r(D)$ . We consider two different cases, depending on whether the minimum enclosing disk  $D$  of the input points is defined by two or more points.



(a) Case 1: Two points define  $D$ . (b) Case 2: At least three points define  $D$ .

■ **Figure 2** Visualization of the two cases in the proof for the existence of a separator that is perpendicular to one of the diameters of  $p_1, p_2, p_3$ .

**Case 1: Two points define  $D$ .** We illustrate this case in Figure 2a. Let  $p_1$  and  $p_2$  be the two points that define the minimum enclosing disk. Then,  $p_1$  must be diametrically opposing  $p_2$  on  $D$ , so  $d(p_1) = d(p_2)$ . As we assume that the optimal solution has value less than  $r(D)$ , we have that  $p_1$  and  $p_2$  must belong to different clusters; without loss of generality, let  $p_1 \in C_1$  and  $p_2 \in C_2$ .

Let  $\ell$  be the line tangent to  $\text{MED}(C_1)$  perpendicular to  $d(p_1)$  furthest in direction  $\overrightarrow{c - p_1}$  (recall that  $c$  is the center of  $D$ ). If  $\ell \cap \text{MED}(C_2) = \emptyset$ , we are done. Otherwise, the projections of the disks  $\text{MED}(C_1)$  and  $\text{MED}(C_2)$  on  $d(p_1)$  cover all of the diameter, so  $r(C_1) + r(C_2) \geq r(D)$ , which is a contradiction.

**Case 2: At least three points define  $D$ .** We illustrate this case in Figure 2b. Let  $p_1, p_2, p_3 \in P$  be any three points that jointly define the minimum enclosing disk. In any optimal solution, two out of these three points must be grouped together in a cluster. Without loss of generality, assume that  $p_1$  and  $p_2$  are in the same cluster.

Any disk containing  $p_1$  and  $p_2$  that is defined by points in  $D$  must contain all of  $\widehat{p_1 p_2}$ , the smallest of the two arcs on  $D$  connecting  $p_1$  and  $p_2$ . We now add the artificial point  $p_3^*$  to the point set – the diametrically opposing point of  $p_3$ . We have that  $p_3^* \in \widehat{p_1 p_2}$  as otherwise  $p_1, p_2, p_3$  would lie strictly inside one half of  $D$  and could therefore not define the minimum enclosing disk by Lemma 3. Hence, adding  $p_3^*$  to the point set does not change the optimal solution. Thus, we reduced our problem to Case 1, where two points define the minimum enclosing disk. ◀

### 3.3 Extension to constant dimensions

In this section, we show how to extend our algorithm for 2-MINSUMRADIUS to any constant dimension  $d$ .

Our main structural lemma in this section is an analog of Lemma 5 generalized to higher dimensions. From this lemma, it follows that we can essentially brute-force the clusters over a small number ( $d + 1$  to be precise) of directions for candidate planes that separate two optimal clusters.

► **Lemma 6.** Consider a point set  $P \subset \mathbb{R}^d$  and let  $Q \subset P$ ,  $|Q| \leq d + 1$ , be a set of points that defines the minimum enclosing ball of  $P$ , i.e.,  $\text{MEB}(Q) = \text{MEB}(P)$ . There exists an optimal 2-MINSUMRADIUS clustering  $C_1, C_2$  of  $P$  such that for a point  $q \in Q$  and a hyperplane  $\ell$  orthogonal to  $d(q)$ , the hyperplane  $\ell$  separates  $C_1$  from  $C_2$ .

Before proving this lemma, let us first consider the algorithm. In order to extend Algorithm 1 so that it works in  $\mathbb{R}^d$ , we need an algorithm to dynamically maintain the minimum enclosing ball of a set of points in  $\mathbb{R}^d$  under insertions and deletions of points. This can be done in sublinear time by a result of Agarwal and Matousek [3]. Namely, the amortized time per update is  $\mathcal{O}(n^{1-1/(\lceil d/2 \rceil + 1) + \epsilon})$ . Since for every candidate direction we need to do at most  $n$  updates (as in the planar case), we have the following result.

► **Theorem 7.** *For a set  $P$  of  $n$  points in  $\mathbb{R}^d$  an optimal 2-MINSUMRADIUS clustering can be computed in  $\mathcal{O}(n^{2-1/(\lceil d/2 \rceil + 1) + \epsilon})$  time.*

**Proof.** Let  $Q \subset P$  be a set of at most  $d + 1$  points that define  $D$ . For each point  $q \in Q$ , there exist  $\mathcal{O}(n)$  combinatorially different hyperplanes that are orthogonal to  $d(q)$  and partition  $P$  into two clusters. For each of these two clusters we have to update the minimum enclosing ball in time  $T(n) = \mathcal{O}(n^{1-1/(\lceil d/2 \rceil + 1) + \epsilon})$  to obtain the 2-MINSUMRADIUS cost. Taking the minimum over all points and all hyperplanes, we find an optimal solution in time  $(d + 1) \cdot n \cdot T(n) = \mathcal{O}(n \cdot T(n))$ . ◀

Let us now prove the structural lemma.

**Proof of Lemma 6.** Analogous to the  $d = 2$  case, assume that the optimal 2-MINSUMRADIUS solution consists of two non-empty clusters, as otherwise an optimal cluster is simply  $\text{MEB}(P)$ . If  $\text{MEB}(P)$  is not an optimal 2-MINSUMRADIUS clustering, then the optimal cost is less than the radius of  $\text{MEB}(P)$  and  $C_1, C_2$  are both non-empty. Let  $Q \subset P, |Q| \leq d + 1$  be a set of points that define the minimum enclosing ball of  $P$ , i.e.,  $\text{MEB}(Q) = \text{MEB}(P)$ , and let  $Q$  lie on the boundary of the minimum enclosing ball of  $P$ , i.e.,  $Q \subset \partial \text{MEB}(P)$ . We argue that a hyperplane orthogonal to a diameter defined by a point in  $Q$  separates  $C_1$  and  $C_2$ . These points can be computed in  $\mathcal{O}(n)$  time [25, 32].

Therefore, consider a partition of  $Q$  into  $Q_1$  and  $Q_2$  and assume without loss of generality that  $Q_1 \subset C_1$  and  $Q_2 \subset C_2$ . Let  $S(P)$  be the minimum enclosing hypersphere of  $P$ , i.e.,  $S(P) = \partial \text{MEB}(P)$ . Let  $S_1 = \text{MEB}(Q_1) \cap S(P)$  and  $S_2 = \text{MEB}(Q_2) \cap S(P)$ , and let  $c$  be the center of  $S(P)$ . Let  $S_1^*$  be  $S_1$  mirrored at  $c$ , i.e., it is all the antipodal points of  $S_1$ . We now consider two cases.

First, suppose that  $S_1^* \cap Q_2 \neq \emptyset$ . Then, we can construct a diameter of  $\text{MEB}(P)$  that intersects both  $S_1$  and  $S_2$ . As there is a point  $q \in Q_2$  with antipodal point  $q^*$  such that  $q \in S_2$  and  $q^* \in S_1$ , both extremes of the diameter  $d(q)$  are contained within a different cluster. As the sum of the radii is at most  $\text{MEB}(P)$ , there is a hyperplane orthogonal to this diameter  $d(q)$  that separates the two clusters.

Second, suppose that  $S_1^* \cap Q_2 = \emptyset$ . We show that this leads to a contradiction. As  $S_1^*$  and  $Q_2$  do not intersect, we can construct a hyperplane  $F$  through  $c$  that separates  $S_1^*$  and  $Q_2$  such that  $(S_1^* \cup Q_2) \cap F = \emptyset$ . Note that as  $F$  contains  $c$  and  $S_1^*$  was mirrored at  $c$ , we have that  $S_1$  and  $Q_2$  are on the same side of  $F$  but do not intersect it. Thus, all points of  $Q = Q_1 \cup Q_2$  are strictly contained on some half of  $\text{MEB}(P)$ . It follows from Lemma 3 that  $\text{MEB}(Q) \neq \text{MEB}(P)$ , which is a contradiction. ◀

## 4 Near-quadratic algorithm for 3-MinSumRadius

In this section, we prove the following result:

► **Theorem 8.** *For a set  $P$  of  $n$  points in the Euclidean plane an optimal 3-MINSUMRADIUS can be computed in expected  $\mathcal{O}(n^2 \log^2 n \log^2 \log n)$  or worst-case  $\mathcal{O}(n^2 \log^4 n)$  time.*



■ **Algorithm 2** Near-quadratic time algorithm for 3-MINSUMRADIUS.

---

```

1: procedure 3-MINSUMRADIUS( $P$ )
2:    $D \leftarrow$  minimum enclosing disk of  $P$ 
3:    $p_1, p_2, p_3 \leftarrow$  points defining  $D$ 
4:    $S_{\text{OPT}} \leftarrow$  2-MINSUMRADIUS( $P$ )
5:   for each diameter  $d(p_i)$  of  $p_i$  do
6:     for each separator  $s$  orthogonal to  $d(p_i)$  through a  $p \in P$  do
7:        $A, B \leftarrow$  sets induced by separator  $s$ , such that  $p_i \in A$ 
8:        $S \leftarrow$  (MED( $A$ ), 2-MINSUMRADIUS( $B$ ))
9:       if  $S$  is better than  $S_{\text{OPT}}$  then
10:         $S_{\text{OPT}} \leftarrow S$ 
11:   return  $S_{\text{OPT}}$ 

```

---

We first give the algorithm in Section 4.1, and then prove the structural result that this algorithm relies on in Section 4.2.

## 4.1 Algorithm

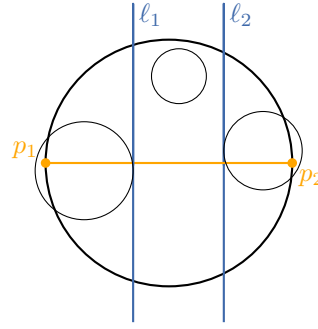
Our near-quadratic algorithm for the 3-MINSUMRADIUS problem again relies on the structural insight that there are only linearly many cuts that need to be considered in order to find one that separates one cluster from the two other clusters. For the separated cluster we can then simply compute the minimum enclosing disk, while for the other two clusters we use our near-linear time algorithm for 2-MINSUMRADIUS. Hence, we obtain a near-quadratic time algorithm.

The following is the main structural lemma that our algorithm relies on:

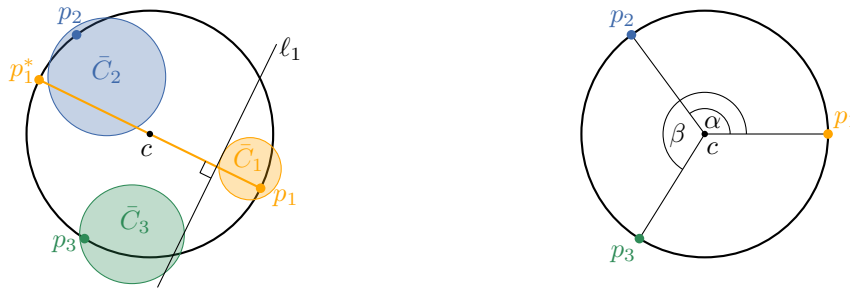
► **Lemma 9.** *Consider a point set  $P$  in the Euclidean plane and let  $p_1, p_2, p_3$  be three points in  $P$  that define the minimum enclosing disk of  $P$  (with potentially  $p_2 = p_3$ ). There exists an optimal 3-MINSUMRADIUS solution  $C_1, C_2, C_3$  to  $P$  such that for a point  $q \in \{p_1, p_2, p_3\}$  and a line  $\ell$  orthogonal to  $d(q)$ , the line  $\ell$  separates the cluster containing  $q$  from the other two clusters.*

We prove this lemma in Section 4.2. Using this lemma, we can now give our full algorithm, see Algorithm 2.

**Algorithm description.** For each point  $p_i$ ,  $i = 1, 2, 3$ , that defines  $D$ , we consider all combinatorially distinct separators that are orthogonal to the diameter  $d(p_i)$ , as shown in Figure 3 for the case with only one distinct possible diameter. For each such separator, let  $A \subset P$  be the set induced by the separator that contains  $p_i$ , and let  $B = P \setminus A$ . We compute the minimum enclosing disk of  $A$  and the 2-MINSUMRADIUS of  $B$  using the algorithm of Theorem 4. Finally, we compare this solution to the best solution found so far, and update it when necessary. Lemma 9 implies that the algorithm will find an optimal clustering this way. As there are only linearly many combinatorially distinct separators orthogonal to a fixed line, and a solution for  $k = 2$  can be computed in expected  $\mathcal{O}(n \log^2 n \log^2 \log n)$  time, the expected running time of the algorithm is  $\mathcal{O}(n^2 \log^2 n \log^2 \log n)$ . The worst-case running time follows by using the deterministic algorithm to maintain the minimum enclosing disk [16].



■ **Figure 3** Visualization of the proof for the existence of a separator that is perpendicular to the diameter of  $p_1$  if the minimum enclosing disk is defined by two points.



■ **Figure 4** Illustration for the notation used in Case 3. Note that to the left,  $\bar{C}_1$  is blocked by  $\bar{C}_3$ .

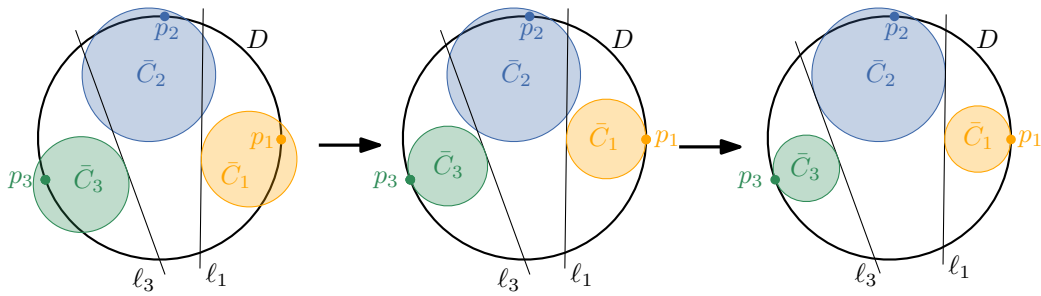
### 4.2 Linear number of cuts

In this subsection, we prove the correctness of our algorithm. That is, we prove Lemma 9 and thus show that it suffices to check a linear number of candidate separators, one of which separates one cluster from the two others.

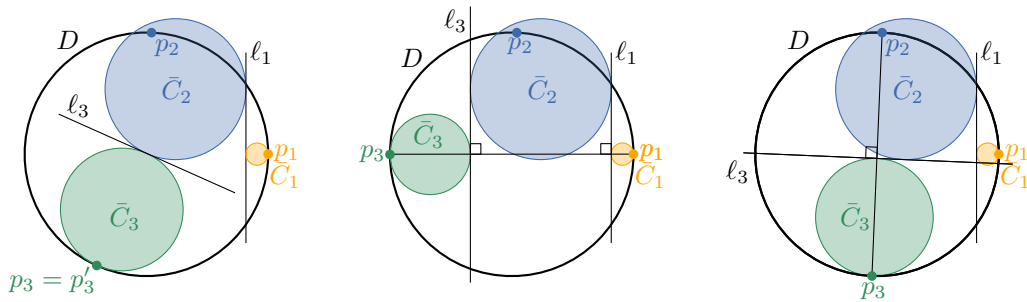
**Proof of Lemma 9.** In this proof, we define  $\bar{C}_i = \text{MED}(C_i)$ . We again assume that  $r(C_1) + r(C_2) + r(C_3) < r(D)$ , as otherwise the solution with a single cluster trivially satisfies our claim. We consider the following cases:

**Case 1: Two points define  $D$ .** Let  $p_1$  and  $p_2$  be the two points that define  $D$ . Then  $p_1$  and  $p_2$  are in different clusters, say  $p_1 \in C_1$  and  $p_2 \in C_2$ . Let  $\ell_i$  be the line perpendicular to  $d(p_1)$  and tangent to  $\bar{C}_i$ , furthest in direction  $\overrightarrow{c - p_i}$ , for  $i \in \{1, 2\}$ . We claim that the lines have the following properties: (i)  $\ell_1 \cap \bar{C}_2 = \emptyset$ , (ii)  $\ell_2 \cap \bar{C}_1 = \emptyset$ , and (iii)  $\ell_1 \cap \bar{C}_3 = \emptyset$  or  $\ell_2 \cap \bar{C}_3 = \emptyset$ . If case (i) or (ii) is false, the projection of the union  $\bar{C}_1 \cup \bar{C}_2$  on the diameter  $d(p_1)$  covers the entire diameter, so the sum of radii is at least  $r(D)$ . If case (iii) is false, the same holds for the union  $\bar{C}_1 \cup \bar{C}_2 \cup \bar{C}_3$  of all the disks. If  $\ell_1 \cap \bar{C}_3 = \emptyset$  then, depending on which side of  $\ell_1$  contains  $C_3$ ,  $\ell_1$  either separates  $C_2$  from  $C_1$  and  $C_3$  or  $C_1$  from  $C_2$  and  $C_3$ . Otherwise, if  $\ell_2 \cap \bar{C}_3 = \emptyset$ , then  $\ell_2$  separates  $C_2$  from  $C_1$  and  $C_3$  or  $C_1$  from  $C_2$  and  $C_3$ .

**Case 2: Three points define  $D$ , two of which are in the same cluster.** Suppose  $D$  is defined by three points  $p_1, p_2, p_3$ , where  $p_1, p_2 \in C_1$ . Then we add an artificial point  $p_3^*$  diametrically opposing  $p_3$  on  $D$ . This reduces this case to Case 1.



■ **Figure 5** The reduction used in case 3a to a situation where  $\bar{C}_1$  and  $\bar{C}_3$  are as small as possible given the points  $p_1, p_3$ .



■ **Figure 6** Situations in case 3a. Left: Placement of  $p_3$  where  $r(C_3)$  is maximum. Middle: Minimum when  $p_1$  and  $p_3$  are diametral. Right: Minimum when  $p_2$  and  $p_3$  are diametral.

**Case 3: Three points define  $D$ , all of which are in different clusters.** Now we consider the final case, where  $p_1 \in C_1, p_2 \in C_2, p_3 \in C_3$ . This is also the most technically involved case. For simplicity, we assume that  $D$  is a unit disk. Define lines  $\ell_1, \ell_2, \ell_3$  as follows: for  $i \in \{1, 2, 3\}$ ,  $\ell_i$  is the line perpendicular to  $d(p_i)$  that is tangent to  $\bar{C}_i$  furthest in the direction  $\overrightarrow{c - p_i}$ ; see Figure 4 for an illustration of our notation. If a line  $\ell_i$  separates  $C_i$  from the other clusters, we are done. We say that a disk  $\bar{C}_j$  blocks a disk  $\bar{C}_i$  if  $\bar{C}_j \cap L_i \neq \emptyset$ , where  $L_i$  is the half-plane bounded by  $\ell_i$  that contains  $\bar{C}_i$ . If  $\ell_i$  is not a separator, it has to be the case that  $L_i$  intersects one of the other two disks. We show that if each disk  $\bar{C}_i$  is blocked by another disk, then  $r(C_1) + r(C_2) + r(C_3) \geq 1$ , which is a contradiction.

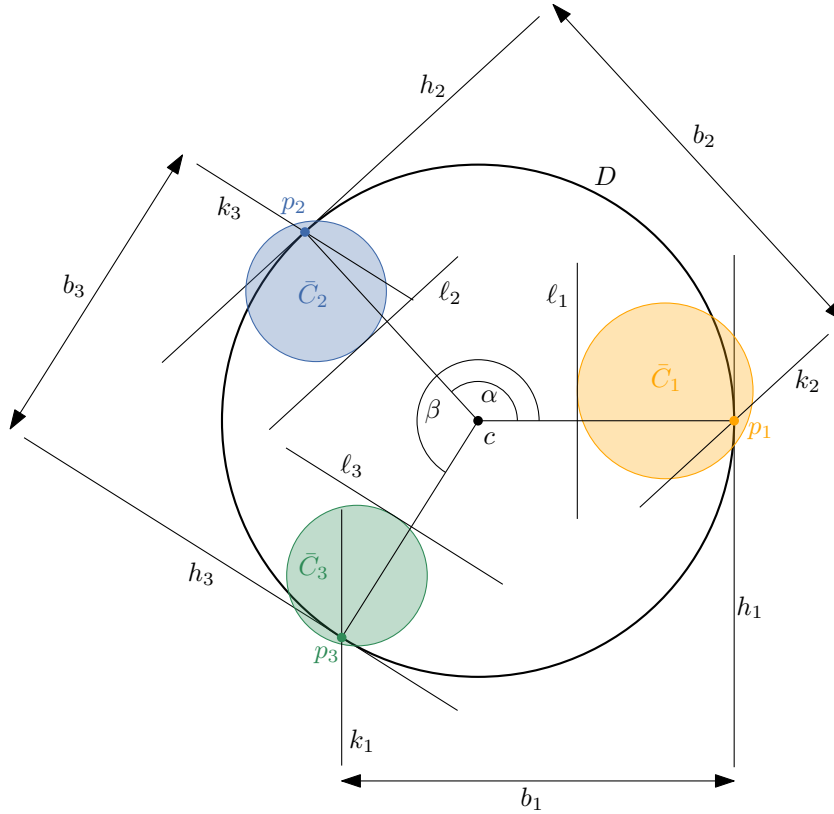
Without loss of generality, assume that the center of  $D$  is  $c = (0, 0)$ , that  $p_1 = (1, 0)$  and that the points appear in the order  $p_1, p_2, p_3$  around  $\partial D$  in counterclockwise direction. Let  $\alpha$  be the angle of  $p_2$  and  $\beta$  be the angle of  $p_3$ . Note that by our assumptions, we have  $\alpha \in (0, \pi]$  and  $\beta \in [\pi, \pi + \alpha]$ . There are two subcases: (3a) there is a disk that blocks both of the other disks, and (3b) the disks block each other in a cyclic pattern.

**Case 3a.** Assume without loss of generality that  $\bar{C}_2$  blocks  $\bar{C}_1$  and  $\bar{C}_3$ . The reductions described in the following are shown in Figure 5. For  $i \in \{1, 3\}$ , consider the disk  $\bar{C}'_i$  that is tangent to  $D$  at  $p_i$  and has the tangent  $\ell_i$ . If not  $\bar{C}_i = \bar{C}'_i$ , then we can replace  $\bar{C}_i$  with  $\bar{C}'_i$ , and this will decrease  $r(C_1) + r(C_2) + r(C_3)$  while maintaining that  $\bar{C}_2$  blocks both of the other disks. After the replacement, the disks no longer contain all input points  $P$ , but we just do it to obtain a lower bound on the sum of radii of the original disks. Hence, we assume that  $\bar{C}_i$  is tangent to  $D$  at  $p_i$ . Furthermore, allowing the tangent  $\ell_i$  to change as well, we can assume that  $\bar{C}_i$  is the smallest disk tangent to  $D$  at  $p_i$  such that  $\bar{C}_2$  blocks  $\bar{C}_i$ , which means that  $\ell_i$  is a separating common tangent of  $\bar{C}_i$  and  $\bar{C}_2$ .

2:12 Clustering with Few Disks to Minimize the Sum of Radii

We now fix  $\bar{C}_1$  and  $\bar{C}_2$  and consider the angle  $\beta \in [\pi, \pi + \alpha]$  of  $p_3$  that causes  $r(C_3)$  to be as small as possible. For a value  $\beta$ , let  $\rho(\beta) = r(C_3)$  be the diameter of the resulting disk  $\bar{C}_3$ . Let  $p'_3$  be point on the boundary of  $D$  on the same diameter as the center of  $\bar{C}_2$  but on the opposite site of the center of  $D$ . The radius  $r(C_3)$  is maximum when  $p_3 = p'_3$ , in which case the two disks  $\bar{C}_2$  and  $\bar{C}_3$  touch; see Figure 6 (left). As  $p_3$  moves away from  $p'_3$  in either direction,  $\bar{C}_3$  shrinks monotonically. It follows that  $\rho$  has no local minima except at one or both of the endpoints  $\beta \in \{\pi, \pi + \alpha\}$ . When  $\beta = \pi$ , the segment  $p_1p_3$  is a diameter of  $D$  and the projection of the disks  $\bar{C}_1, \bar{C}_2, \bar{C}_3$  on  $p_1p_3$  covers the entire diameter, so  $r(C_1) + r(C_2) + r(C_3) \geq 1$ ; see Figure 6 (middle). When  $\beta = \pi + \alpha$ , the segment  $p_2p_3$  is a diameter of  $D$  and the projection of the disks  $\bar{C}_2, \bar{C}_3$  on  $p_2p_3$  covers the entire diameter, so  $r(C_1) + r(C_2) + r(C_3) \geq r(C_2) + r(C_3) \geq 1$ ; see Figure 6 (right).

**Case 3b.** Without loss of generality, assume that  $\bar{C}_i$  blocks  $\bar{C}_{i+1}$  for each  $i \in \{1, 2, 3\}$  (with indices taken modulo 3). Let  $h_i$  be the tangent to  $D$  in  $p_i$  and  $k_i$  be the line through  $p_{i-1}$  parallel to  $h_i$ ; see Figure 7. Then the three lines  $h_i, \ell_i$  and  $k_i$  are all parallel.



■ **Figure 7** The setting in case 3b.

Since  $\bar{C}_i$  blocks  $\bar{C}_{i+1}$ , the sum  $2r(C_i) + 2r(C_{i+1})$  is at least the distance between the lines  $h_{i+1}$  and  $k_{i+1}$ . Define  $b_{i+1}$  to be this distance. We then have  $4(r(C_1) + r(C_2) + r(C_3)) \geq b_1 + b_2 + b_3$ , and we aim at showing  $b_1 + b_2 + b_3 \geq 4$ .

We first fix  $p_2$  and analyze the placement of  $p_3$  that results in the smallest value of  $b_3 + b_1$ . In other words, we fix  $\alpha$  and consider the value of  $\beta \in [\pi, \pi + \alpha]$  as a free variable. We observe that  $b_3 = 1 - \cos(\beta - \alpha)$  and  $b_1 = 1 - \cos(2\pi - \beta) = 1 - \cos \beta$ . Hence,

$$b_3 + b_1 = 2 - \cos(\beta - \alpha) - \cos \beta = 2 - \cos((\beta - \alpha/2) - \alpha/2) - \cos((\beta - \alpha/2) + \alpha/2).$$

The sum formula  $\cos(x + y) = \cos x \cos y - \sin x \sin y$  gives

$$b_3 + b_1 = 2 - 2 \cos(\alpha/2) \cos(\beta - \alpha/2).$$

Now it is clear that  $b_3 + b_1$  is maximum when  $\beta = \pi + \alpha/2$  and minimum when  $\beta \in \{\pi, \pi + \alpha\}$ . For both of these minimum values, we get

$$b_3 + b_1 = 2 + 2 \cos^2(\alpha/2).$$

Note that  $b_2 = 1 - \cos \alpha$ , so we get

$$b_1 + b_2 + b_3 = 1 - \cos \alpha + 2 + 2 \cos^2(\alpha/2) = 3 - \cos \alpha + 2 \cos^2(\alpha/2).$$

By the identity  $\cos(2x) = 2 \cos^2 x - 1$ , we get  $b_1 + b_2 + b_3 = 4$ , and we have the desired contradiction.  $\blacktriangleleft$

## 5 Conclusion

In this work, we presented two new clustering algorithms for point sets in the plane: one for 2-MINSUMRADIUS and one for 3-MINSUMRADIUS. We improve the running time for the former problem from near-quadratic to near-linear, and generalize the algorithm to a sub-quadratic algorithm for any constant dimension. For the latter problem we give the first non-naive algorithm, matching the running time of the previously best algorithm for 2-MINSUMRADIUS. At the heart of our algorithms are new structural insights on separators of optimal solutions, which might be of independent interest. We showed that there are only linearly many separators that need to be considered for  $k = 2$  as well as  $k = 3$ . Hence, it is conceivable that there also exists a near-linear time algorithm for 3-MINSUMRADIUS.

An interesting direction for future work is to understand whether any of our structural insights can be pushed further to obtain a significantly improved running time for general  $k$ -MINSUMRADIUS clustering. Another natural future line of work is to extend our algorithm for 3-MINSUMRADIUS to higher dimensions, which could be another step towards an efficient algorithm for  $k$ -MINSUMRADIUS in higher dimensions. To understand the inherent hardness of the problems considered in this work, it would be of interest to show fine-grained lower bounds for some combinations of the ambient dimension and  $k$ , similar to [11].

We may also consider the *rectilinear*  $k$ -MINSUMRADIUS problem, where we aim to cover a set of points using  $k$  axis-parallel squares of minimum total side length. For  $k = 2$ , this problem has a straight-forward  $\mathcal{O}(n \log n)$  time algorithm, which is optimal in the algebraic computation tree model by a reduction from the Maximum Gap problem [23]. For  $k = 3$ , it is easy to see that there must be a vertical or horizontal line that separates one square from the other two in an optimal solution, so we obtain an  $\mathcal{O}(n^2 \log n)$  time algorithm. It is interesting whether a faster algorithm exists. Again, the analogous  $k$ -CENTER problem of covering a set of points with  $k$  squares of minimum maximum size has received much more attention, and  $\mathcal{O}(n)$  time algorithms have been described for up to three clusters [15, 21, 30] and  $\mathcal{O}(n \log n)$  time algorithms for up to five clusters [27, 28, 30].

---

## References

- 1 Mikkel Abrahamsen, Anna Adamaszek, Karl Bringmann, Vincent Cohen-Addad, Mehran Mehr, Eva Rotenberg, Alan Roytman, and Mikkel Thorup. Fast fencing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 564–573. ACM, 2018. doi:10.1145/3188745.3188878.

- 2 Mikkel Abrahamsen, Mark de Berg, Kevin Buchin, Mehran Mehr, and Ali D. Mehrabi. Minimum perimeter-sum partitions in the plane. *Discret. Comput. Geom.*, 63(2):483–505, 2020. doi:10.1007/s00454-019-00059-0.
- 3 Pankaj K. Agarwal and Jiri Matousek. Dynamic half-space range reporting and its applications. *Algorithmica*, 13(4):325–345, 1995. doi:10.1007/BF01293483.
- 4 Pankaj K. Agarwal and Micha Sharir. Planar geometric location problems. *Algorithmica*, 11(2):185–195, 1994. doi:10.1007/BF01182774.
- 5 Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. NP-hardness of Euclidean sum-of-squares clustering. *Mach. Learn.*, 75(2):245–248, 2009. doi:10.1007/s10994-009-5103-0.
- 6 Helmut Alt, Esther M. Arkin, Hervé Brönnimann, Jeff Erickson, Sándor P. Fekete, Christian Knauer, Jonathan Lenchner, Joseph S. B. Mitchell, and Kim Whittlesey. Minimum-cost coverage of point sets by disks. In *Proceedings of the 22nd ACM Symposium on Computational Geometry, SoCG*, pages 449–458. ACM, 2006. doi:10.1145/1137856.1137922.
- 7 Mihai Badoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing, STOC*, pages 250–257. ACM, 2002. doi:10.1145/509907.509947.
- 8 Moritz Buchem, Katja Ettmayr, Hugo Kooki Kasuya Rosado, and Andreas Wiese. A  $(3+\epsilon)$ -approximation algorithm for the minimum sum of radii problem with outliers and extensions for generalized lower bounds. *CoRR*, abs/2311.06111, 2023. doi:10.48550/ARXIV.2311.06111.
- 9 Vasilis Capoyleas, Günter Rote, and Gerhard J. Woeginger. Geometric clusterings. *J. Algorithms*, 12(2):341–356, 1991. doi:10.1016/0196-6774(91)90007-L.
- 10 Timothy M. Chan. More planar two-center algorithms. *Comput. Geom.*, 13(3):189–198, 1999. doi:10.1016/S0925-7721(99)00019-X.
- 11 Timothy M. Chan, Qizheng He, and Yuancheng Yu. On the fine-grained complexity of small-size geometric set cover and discrete  $k$ -center for small  $k$ . In *50th International Colloquium on Automata, Languages, and Programming, ICALP*, volume 261 of *LIPICs*, pages 34:1–34:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.ICALP.2023.34.
- 12 Moses Charikar and Rina Panigrahy. Clustering to minimize the sum of cluster diameters. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, STOC*, pages 1–10. ACM, 2001. doi:10.1145/380752.380753.
- 13 Kyungjin Cho and Eunjin Oh. Optimal algorithm for the planar two-center problem. *CoRR*, abs/2007.08784, 2020. arXiv:2007.08784.
- 14 Sanjoy Dasgupta and Yoav Freund. Random projection trees for vector quantization. *IEEE Trans. Inf. Theory*, 55(7):3229–3242, 2009. doi:10.1109/TIT.2009.2021326.
- 15 Zvi Drezner. On the rectangular  $p$ -center problem. *Naval Research Logistics (NRL)*, 34(2):229–234, 1987.
- 16 David Eppstein. Dynamic three-dimensional linear programming. *INFORMS J. Comput.*, 4(4):360–368, 1992. doi:10.1287/ijoc.4.4.360.
- 17 David Eppstein. Faster construction of planar two-centers. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 131–138. ACM/SIAM, 1997. URL: <http://dl.acm.org/citation.cfm?id=314161.314198>.
- 18 Matt Gibson, Gaurav Kanade, Erik Krohn, Imran A. Pirwani, and Kasturi R. Varadarajan. On clustering to minimize the sum of radii. *SIAM J. Comput.*, 41(1):47–60, 2012. doi:10.1137/100798144.
- 19 Johann Hagauer and Günter Rote. Three-clustering of points in the plane. *Comput. Geom.*, 8:87–95, 1997. doi:10.1016/S0925-7721(96)00022-3.
- 20 John Hershberger. Minimizing the sum of diameters efficiently. *Comput. Geom.*, 2:111–118, 1992. doi:10.1016/0925-7721(92)90028-Q.

- 21 Michael Hoffmann. A simple linear algorithm for computing rectilinear 3-centers. *Comput. Geom.*, 31(3):150–165, 2005. doi:10.1016/J.COMGEO.2004.12.002.
- 22 Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted voronoi diagrams and randomization to variance-based  $k$ -clustering. In *Proceedings of the 10th Annual Symposium on Computational Geometry, SoCG*, pages 332–339. ACM, 1994. doi:10.1145/177424.178042.
- 23 D. T. Lee and Ying-Fung Wu. Geometric complexity of some location problems. *Algorithmica*, 1(2):193–211, 1986. doi:10.1007/BF01840442.
- 24 Meena Mahajan, Prajakta Nimbhorkar, and Kasturi R. Varadarajan. The planar  $k$ -means problem is NP-hard. *Theor. Comput. Sci.*, 442:13–21, 2012. doi:10.1016/j.tcs.2010.05.034.
- 25 Nimrod Megiddo. Linear-time algorithms for linear programming in  $\mathbb{R}^3$  and related problems. *SIAM J. Comput.*, 12(4):759–776, 1983. doi:10.1137/0212052.
- 26 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.
- 27 Doron Nussbaum. Rectilinear  $p$ -piercing problems. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC*, pages 316–323, 1997. doi:10.1145/258726.258828.
- 28 Michael Segal. On piercing sets of axis-parallel rectangles and rings. In *Proceedings of the Annual European Symposium on Algorithms, ESA 1997*, pages 430–442, 1997. doi:10.1007/3-540-63397-9\_33.
- 29 Micha Sharir. A near-linear algorithm for the planar 2-center problem. *Discret. Comput. Geom.*, 18(2):125–134, 1997. doi:10.1007/PL00009311.
- 30 Micha Sharir and Emo Welzl. Rectilinear and polygonal  $p$ -piercing and  $p$ -center problems. In *Symposium on Computational Geometry (SoCG 1996)*, pages 122–132, 1996. doi:10.1145/237218.237255.
- 31 Haitao Wang. On the planar two-center problem and circular hulls. *Discret. Comput. Geom.*, 68(4):1175–1226, 2022. doi:10.1007/s00454-021-00358-5.
- 32 Emo Welzl. Smallest enclosing disks (balls and ellipsoids). In *New Results and New Trends in Computer Science, Proceedings [on occasion of H. Maurer’s 50th birthday]*, volume 555 of *Lecture Notes in Computer Science*, pages 359–370. Springer, 1991. doi:10.1007/BFB0038202.