Clustering with Few Disks to Minimize the Sum of Radii*

Mikkel Abrahamsen[1], Sarita de Berg[2], Lucas Meijer[2], André Nusser[3], and Leonidas Theocharous[4]

1 **University of Copenhagen, Denmark**
miab@di.ku.dk

2 **Utrecht University, The Netherlands**
s.deberg@uu.nl, l.meijer2@uu.nl

3 **CNRS, Inria, I3S, Université Côte d'Azur, France**
andre.nusser@inria.fr

4 **Eindhoven University of Technology, The Netherlands**
l.theocharous@tue.nl

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. 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 $\mathcal{O}(n^{2-1/(\lceil d/2 \rceil+1)+\varepsilon})$ 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.

Related Version Full version: https://arxiv.org/abs/2312.08803

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 [6]. In their general form, as defined in [6], 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.

* MA 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. LM is supported by the Netherlands Organisation for Scientific Research (NWO) under project no. VI.Vidi.213.150. AN was part of Basic Algorithms Research Copenhagen (BARC), supported by the VILLUM Foundation grant 16582 during part of this work. LT is supported by the Dutch Research Council (NWO) through Gravitation-grant NETWORKS-024.002.003. This work was initiated at the Workshop on New Directions in Geometric Algorithms, May 14–19 2023, Utrecht, The Netherlands.

40th European Workshop on Computational Geometry, Ioannina, Greece, March 13–15, 2024.
This is an extended abstract of a presentation given at EuroCG'24. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

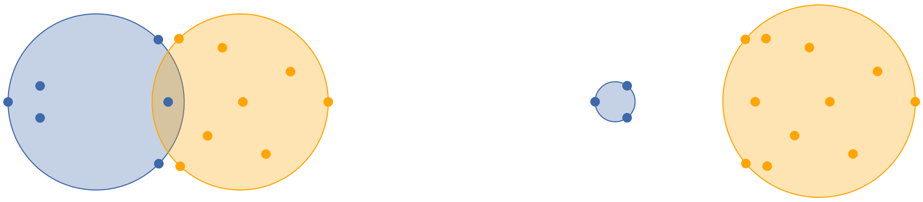


Figure 1 The optimal 2-Center clustering (left) compared to the optimal 2-MinSumRadius clustering (right) for the same point set. In this example 2-MinSumRadius clustering better captures the structure of the point set than 2-Center clustering.

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 (see Figure 1), 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 [14, 16], the k-MinSumRadius problem can be solved in polynomial time [12]: $\mathcal{O}(n^{881})$ [1]. Although 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 and as clustering into a small number of clusters is practically more relevant, we consider the k-MinSumRadius problem for small values of k here. 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 [3, 4, 7, 8, 9, 11, 13, 17, 18] — also called *bi-partition*. Bi-partition problems are 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 [7, 11, 8, 17, 18], the best known algorithm for 2-MinSumRadius still has near-quadratic expected running time $\mathcal{O}(n^2 \log^2 n \log^2 \log n)$ [10], which has seen no improvement in the last 30 years, despite significant work on related problems.

[1] 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.

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 $\mathcal{O}(n^{2-1/(\lceil d/2 \rceil+1)+\varepsilon})$. 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, 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 [10], which increases the running times to $O(n \log^4 n)$ and $O(n^2 \log^4 n)$, 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 these 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 [10], which considered quadratically many separators. We then check all clusterings induced by these separators using an algorithm from [10] 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 $d \geq 3$ [2]. 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 $k \in \{2, 3\}$.

Structure of the paper. Due to space limitations, in this paper we focus on presenting the algorithm and correctness of the planar $k = 2$ case, in Section 2 and Section 3. We give a brief overview of the $k = 3$ case in Section 4. As outlined above, the algorithms for the other cases than planar $k = 2$ are similar and rely on an analogous structural result for their respective case, namely that we can identify a linear number of separators out of which one separates one of the optimal clusters from the rest of the clusters. The extension to constant dimensions for the $k = 2$ case as well as the proof for the $k = 3$ case — which is significantly more technical — can be found in the full version [1].

2 Preliminaries

For a set of points $Q \subset \mathbb{R}^2$, let MED(Q) be the minimum enclosing disk that contains all points of Q and let $r(Q)$ be the radius of MED(Q).

▶ **Definition 2.1** (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 $C_1, C_2, ..., C_k$ such that $\sum_{i=1}^{k} r(C_i)$ is minimized.

Throughout the paper, let D denote the minimum enclosing disk of the input points of our k-MinSumRadius instance, i.e., $D := \text{MED}(P)$, and let c be the center of D. We say that a point set Q defines a disk D' if $\text{MED}(Q) = D'$. Let p be a point on the boundary of D, we define p^* to be the diametrically-opposing point of p on D. The diameter $d(p)$ is then 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 disks are disjoint.

▶ **Lemma 2.2.** *Given a k-MinSumRadius instance, there exists an optimal solution with clusters $C_1, \dots, C_k$ such that $MED(C_i) \cap MED(C_j) = \emptyset$ for all distinct $i, j \in [k]$.*

Proof. Consider an arbitrary optimal solution for which there are two clusters C_i, C_j where $\mathrm{MED}(C_i) \cap \mathrm{MED}(C_j) \neq \emptyset$. We 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 cost, as $r(C') \leq r(C_i) + r(C_j)$. Hence, we obtain a clustering with one less 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 disks. ◀

3 Near-Linear Algorithm for 2-MinSumRadius

In this section, we present our algorithm for 2-MinSumRadius for the plane. We generalize this to higher dimensions in the full version of this paper. For the plane, we prove the following result:

▶ **Theorem 3.1.** *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 $O(n \log^4 n)$ time.*

3.1 Algorithm

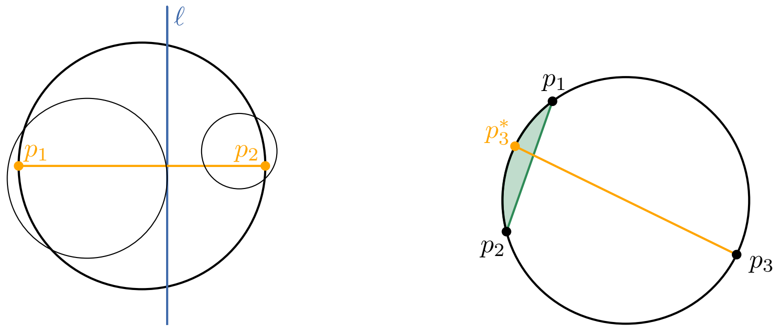
Our algorithm uses the insight that there exist a linear number of separators, such that one of them separates the points in cluster C_1 from those in cluster C_2.

▶ **Lemma 3.2.** *Given a point set P in the Euclidean plane, let C_1, C_2 be an optimal 2-MinSumRadius clustering of P. Furthermore, let p_1, p_2, p_3 be three points of P that define the minimum enclosing disk of P (with potentially $p_2 = p_3$). Then there exists a point $q \in \{p_1, p_2, p_3\}$ and a line ℓ orthogonal to $d(q)$ such that ℓ separates C_1 from C_2.*

We prove this result in Section 3.2. We now explain our algorithm relying on Lemma 3.2.

Algorithm description. Let p_1, p_2, p_3 denote a triple of points in P that define D (possibly $p_2 = p_3$). These points can be computed in $O(n)$ time [15]. 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 or multiple collinear points switch sides with respect to the separator. This ensures that the minimum enclosing disk on one side of the separator is incremental while it is decremental on the other side. 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 3.2, 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 $O(\log^4 n)$ amortized time per update [10]. So, checking all n separators requires expected $\mathcal{O}(n \log^2 n \log^2 \log n)$ or $O(n \log^4 n)$ worst-case time. As we only have diameters $d(p_1)$, $d(p_2)$, and $d(p_3)$ to handle, we can compute the optimal solution in the same time.



(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.

3.2 Linear number of cuts

In this subsection we prove Lemma 3.2, which states that it suffices to only check separators orthogonal to one of the diameters $d(p_1)$, $d(p_2)$, or $d(p_3)$. From now on, we assume that the optimal solution consists of two non-empty clusters. If the optimal solution contains only a single cluster, it must be D. So, only solutions that have sum of radii strictly smaller than $r(D)$ are relevant for us in the case of two non-empty clusters.

Proof of Lemma 3.2. We consider two different cases, depending on whether the minimum enclosing disk D of the input points is defined by two or more points.

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 ℓ be the line tangent to $\mathrm{MED}(C_1)$ perpendicular to $d(p_1)$ furthest in direction $\overrightarrow{c - p_1}$ (recall that c is the center of D). If ℓ does not separate $\mathrm{MED}(C_1)$ and $\mathrm{MED}(C_2)$, then the projections of the disks $\mathrm{MED}(C_1)$ and $\mathrm{MED}(C_2)$ on $d(p_1)$ cover all of the diameter, so $r(C_1) + r(C_2) \geq r(D)$, which is a contradiction. Hence, ℓ separates the clusters as desired.

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 (thus, also has radius at most $r(D)$) must contain all of $\widehat{p_1p_2}$, 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 $p_3^* \in \widehat{p_1p_2}$ 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 [5, Lemma 2.2]. 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. ◀

4 Near-Quadratic Algorithm for 3-MinSumRadius

Due to the space limit, we can only give a very brief description of the $k = 3$ case here and we refer to the full version for proofs and more details. 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 4.1.** *Given a point set P in the Euclidean plane, let C_1, C_2, C_3 be an optimal 3-MinSumRadius clustering of P. Furthermore, let p_1, p_2, p_3 be three points in P that define the minimum enclosing disk of P (with potentially $p_2 = p_3$). Then there exists a point $q \in \{p_1, p_2, p_3\}$ and a line ℓ orthogonal to $d(q)$ such that ℓ separates the cluster containing q from the other two clusters.*

The proof can be found in the full version. We prove this lemma by a case distinction on which cluster the points that define the minimum enclosing disk D belong to. We then either reduce to the case in which the endpoints of one of the diameters lie in different clusters (similar to the k = 2 case), or we show that non-existence of a separator results in a contradiction.

We then obtain the following theorem.

▶ **Theorem 4.2.** *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.*

References

1 Mikkel Abrahamsen, Sarita de Berg, Lucas Meijer, André Nusser, and Leonidas Theocharous. Clustering with few disks to minimize the sum of radii. *CoRR*, abs/2312.08803, 2023. arXiv:2312.08803, doi:10.48550/ARXIV.2312.08803.

2 Pankaj K. Agarwal and Jirí Matousek. Dynamic half-space range reporting and its applications. *Algorithmica*, 13(4):325–345, 1995. doi:10.1007/BF01293483.

3 Pankaj K. Agarwal and Micha Sharir. Planar geometric location problems. *Algorithmica*, 11(2):185–195, 1994. doi:10.1007/BF01182774.

4 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.

5 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.

6 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.

7 Timothy M. Chan. More planar two-center algorithms. *Comput. Geom.*, 13(3):189–198, 1999. doi:10.1016/S0925-7721(99)00019-X.

8 Kyungjin Cho and Eunjin Oh. Optimal algorithm for the planar two-center problem. *CoRR*, abs/2007.08784, 2020. arXiv:2007.08784.

9 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.

10 David Eppstein. Dynamic three-dimensional linear programming. *INFORMS J. Comput.*, 4(4):360–368, 1992. doi:10.1287/ijoc.4.4.360.

11 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.

12 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.

13 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.

14 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.

15 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.

16 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.

17 Micha Sharir. A near-linear algorithm for the planar 2-center problem. *Discret. Comput. Geom.*, 18(2):125–134, 1997. doi:10.1007/PL00009311.

18 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.