Improved Fixed-Parameter Bounds for Min-Sum-Radii and Diameters k-Clustering and Their Fair Variants

Sandip Banerjee[1], **Yair Bartal**[2], **Lee-Ad Gottlieb**[3], **Alon Hovav**[2]
[1]IDSIA USI-SUPSI, Switzerland
[2]The Hebrew University of Jerusalem, Israel
[3]Ariel University, Israel
sandip.ndp@gmail.com, yair@cs.huji.ac.il, leead@ariel.ac.il, alon.hovav@mail.huji.ac.il

Abstract

We provide improved upper and lower bounds for the Min-Sum-Radii (MSR) and Min-Sum-Diameters (MSD) clustering problems with a bounded number of clusters k. In particular, we propose an exact MSD algorithm with running-time $n^{O(k)}$. We also provide $(1+\epsilon)$ approximation algorithms for both MSR and MSD with running-times of $O(kn) + (1/\epsilon)^{O(dk)}$ in metrics spaces of doubling dimension d. Our algorithms extend to k-center, improving upon previous results, and to α-MSR, where radii are raised to the α power for $\alpha > 1$. For α-MSD we prove an exponential time ETH-based lower bound for $\alpha > \log 3$. All algorithms can also be modified to handle outliers. Moreover, we can extend the results to variants that observe *fairness* constraints, as well as to the general framework of *mergeable* clustering, which includes many other popular clustering variants. We complement these upper bounds with ETH-based lower bounds for these problems, in particular proving that $n^{O(k)}$ time is tight for MSR and α-MSR even in doubling spaces, and that $2^{o(k)}$ bounds are impossible for MSD.

1 Introduction

In this paper, we consider two basic clustering problems, both of which are well-studied and the subject of very recent interest. These are the **min-sum radii** (MSR) and **min-sum diameters** (MSD) clustering problems. For these problems, the input is a set of points equipped with a metric distance function along with an integral parameter k. The task is to partition the points into k clusters, while minimizing the sum of cluster radii or diameters, respectively.

These problems have been the subject of study for several decades (Brucker 1978; Hansen and Jaumard 1987; Monma and Suri 1991; Capoyleas, Rote, and Woeginger 1991), and so it is unsurprising that several of their natural variants have received significant attention in the literature as well. A simple yet challenging one among these is the *outliers* variant, where a solution need only cover $n-g$ of the n input points (Buchem et al. 2024), where g is the number of outliers. A second, more profound variant is the α version – that is **α-MSR** and **α-MSD** – wherein the objective function is sum of the α power of the radius or diameter (where $\alpha > 1$) (Charikar and Panigrahy 2001; Bandyapadhyay and Varadarajan 2016). Additional important variants of these problems, which incorporate various *fairness* constraints, have garnered much recent interest (Arutyunova and Schmidt 2021; Drexler et al. 2023; Chen et al. 2024). Many of these variants, as well as clustering with lower bound constraints, are captures within the framework of *mergeable* clustering (Arutyunova and Schmidt 2021; Drexler et al. 2023).

In this paper, we consider the general metric setting, and continue in a line of research which studies these problems under the popular *fixed parameter tractable* (FPT) model, wherein k is taken as fixed (see, for example, (Behsaz and Salavatipour 2015; Bandyapadhyay, Lochet, and Saurabh 2023; Chen et al. 2024)). Algorithms under this model may achieve superior run-time dependence on n, at the cost of a steep (typically exponential) dependence on parameter k. We seek both exact and approximate algorithms for MSR and MSD and their variants, and improve upon many previous FPT results. Our algorithms provide fixed parameter polynomial time approximation schemes (PTAS) – that is $(1+\epsilon)$-approximations with run-time polynomial in n and dependent on k and ϵ – for all these problems.

Our results for general metric spaces and fixed k (summarized in Tables 1,2 and 3) are as follows:

Exact Algorithms. For MSR, it is easy to see that a brute-force algorithm solves the problem in time $n^{O(k)}$. This algorithm can also be used for the case of g outliers, or when the distance is raised to the power $\alpha > 1$. Our first contribution is showing that the naive algorithm is in fact optimal: Assuming that the Exponential Time Hypothesis (ETH) holds, MSR cannot be solved in time $n^{o(k)}$. This lower-bound holds for α-MSR and in the presence of outliers as well.

For MSD, we improve upon the algorithm presented in (Behsaz and Salavatipour 2015) with run-time $n^{O(k^2)}$, proving that it can be modified to create clusters in increasing order of diameter, as the intersection at most of a constant number of balls (rather than k in the original algorithm). This allows us to match for metric MSD the run-time of $n^{O(k)}$ previously known only for metric MSR and Euclidean MSD (Capoyleas, Rote, and Woeginger 1991). Here too we can handle outliers without increasing the run time. See Table 1. In terms of hardness, we show that assuming ETH, MSD does not admit algorithms with run-time $2^{o(k)}$. For

α-MSD, we show that ETH rules out a run-time of $n^{o(k)}$ for $\alpha \in (1, \log_2 3]$ and $2^{o(n)}$ for $\alpha > \log_2 3$. See Table 2.

Approximation Algorithms. We consider $(1+\epsilon)$-approximation algorithms for a wide of range of settings, also in the presence of outliers. We also give approximation algorithms for α-MSR (again, also in the presence of outliers), while ruling out a PTAS for α-MSD (See Table 2). The upper bound can be viewed as an approximation for the l_α-norm of the radii version and in particular for the k-center problem, in which the cost of the solution is the maximum radius over all solution ball, improving upon existing results (Feldmann and Marx 2018).

Our approximation run-times all feature exponential dependence on the *doubling dimension* (denoted by d), a widely used measure of the growth rate of metric space. We recall that by definition $d \leq \log n$ – so that our results give PTAS for all values of d. The assumption that the dimension is low is reasonable in applied settings which allow efficient dimensionality reduction (Bartal, Fandina, and Neiman 2022).

The initial step for all these algorithms is a decomposition method, similar to the technique of (Banerjee et al. 2024). This partitions the original problem into individual sub-problems – each sub-problem has bounded aspect ratio and diameter within a fixed factor of the optimal cost of the original problem. This enables us to consider ϵ-nets of only limited fineness for each sub-problem, and in turn allows us to bound the number of points in each net using the doubling dimension of the space. The solutions on the ϵ-nets of the individual sub-problems are computed and then merged into a single solution for the original problem.

For each sub-problem, the algorithm first enumerates over a bounded set of guesses for estimates of the optimal cost of the sub-problem. For each such choice, it builds candidate solutions in a recursive manner: Given a previously computed partial solution, we iterate over all points contained in yet undiscovered clusters, and for each one compute a candidate covering cluster containing it. The bounded aspect ratio of the space together with the bound on total cost allows us to bound the number of candidate radii for each candidate cluster. This process becomes more intricate for MSD and its variants, where the choice of the point and the task of creating a valid cluster are much more involved.

For both MSR and MSD, we give $(1+\epsilon)$-approximate solutions in time $\left(\frac{1}{\epsilon}\right)^{O(kd)} + \min\{O(kn), 2^{O(d)} n \log n\}$, where the second term is due to the decomposition step, and the first to the algorithmic run on the individual sub-problems. The presence of outliers adds a factor of $g^{O(d)}$ to the first term in the MSR and MSD run-times, and for MSD we also have an additional factor of $\binom{k+g}{g}$. For α-MSR, the dependence on $\frac{1}{\epsilon}$ is replaced with $\frac{\alpha}{\epsilon}$. See Table 3.

Extension to Fairness and Mergeable Clustering. The notion of *fairness* describes a solution which is intuitively balanced. Many different notions of fairness have been suggested for various clustering problems. (Chen et al. 2024) defined a fair version of MSR, wherein we are given as input a coloring of the points, and require that at most a predetermined number of centers may be chosen from each color.

FPT exact algorithms for metric MSD

	Run-time	Ref.	Previous
MSD	$n^{O(k)}$	Thm 3	$n^{O(k^2)}$
MSD with outliers	$n^{O(k)}$	Thm 4	-
Fair MSD	$n^{O(k)}$	Thm 11	-

Table 1: Exact algorithms for MSD. Previous result is due to (Behsaz and Salavatipour 2015).

ETH hardness results

	Run-time	Notes
MSR & α-MSR	$n^{\Omega(k)}$ & $n^{\Omega(\log(\Phi))}$	even when $d = O(1)$
MSD	$2^{\Omega(k)}$	even for PTAS in Euclidean
α-MSD	$n^{\Omega(k)}$	$\forall \alpha > 1$, even when $d = O(1)$
	$2^{\Omega(n)}$	$\forall \alpha > \log_2 3$ and $k \geq 3$

Table 2: Hardness for exact algorithms. The second and third items hold also for PTAS. The first bound holds for any $k \leq n^{1-o(n)}$. Proofs are given in the full paper.

We obtain for this Fair MSR problem the same exact and approximation results as for regular MSR. Previously, only a $(3+\epsilon)$-approximation was known for general metric space.

The above notion of fairness does not apply to MSD, where clusters are not centered at points. Instead we consider the definition introduced in (Arutyunova and Schmidt 2021), who defined a class of *mergeable* clustering problems. A clustering problem is mergeable if for every valid solution, merging any two clusters in this solution will again yield a valid solution. It was shown that various popular clustering constraints (including interesting several variants of fairness (Drexler et al. 2023) , and lower bound constraints (Arutyunova and Schmidt 2021)), are all particular cases of mergeable clustering. For mergeable MSD we give a $(1+\epsilon)$-approximation in time $\left(\frac{1}{\epsilon}\right)^{O(kd)} 2^{O(k/\epsilon)} + 2^{O(d)} n \log n + O(kn)$.

Related work

MSR: This problem is known to be NP-hard even for metrics of constant doubling dimension, as well as for metrics induced by weighted planar graphs (Gibson et al. 2010). In the metric setting, an exact algorithm with quasi-polynomial run-time $n^{O(\log n \log \Phi)}$ is known (where Φ is the aspect ratio of the points) (Gibson et al. 2010), as is a related run-time of $n^{O(d \log d + \log \Phi)}$ (Banerjee et al. 2024), where $d \leq \log n$ is the doubling dimension. For Euclidean space, a recent result gives a run-time of $n^{O(d \log d)}$ (Banerjee et al. 2024) for the discrete problem (where d is the Euclidean dimension).

A line of work has recently yielded a $(3+\epsilon)$-approximate algorithm for the metric problem (Charikar and Panigrahy 2001; Friggstad and Jamshidian 2022; Buchem et al. 2024; Bandyapadhyay, Lochet, and Saurabh 2023). In (Banerjee et al. 2024) we have recently provided a $(1+\epsilon)$-approximation in time $O(kn) + k^{O_\epsilon(d)} (\log k)^{\tilde{O}(d^2)}$. Our results here improve upon this bound when $k = o(d)$, reducing

Approximation Algorithms

	Algorithm run time	Decomposition run time	Ref.	Previous results Approx.	Time	Setting
MSR	$\left(\frac{1}{\epsilon}\right)^{O(kd)}$	$O(kn)$	Thm 8	$(1+\epsilon)$	$2^{O(kd\log(k/\epsilon))}n^3$	Euclidean
		or		$(2+\epsilon)$	$2^{(k\log k/\epsilon)}n^3$	Metric
MSD	$\left(\frac{1}{\epsilon}\right)^{O(kd)}$	$2^{O(d)}n\log n$	Thm 9	$(6+\epsilon)$	$n^{O(1/\epsilon)}$	Metric
α-MSR	$\left(\frac{\alpha}{\epsilon}\right)^{O(kd)}$		*	-		
k-center	$\left(\frac{1}{\epsilon}\right)^{O(kd)}$		*	$(1+\epsilon)$	$n^2k^k\left(\frac{1}{\epsilon}\right)^{O(kd)}$	Metric
MSR outliers	$g^{O(d)}\binom{k+g}{g}\left(\frac{1}{\epsilon}\right)^{O(kd)}$	$O((k+g)n)$	*	$(3+\epsilon)$	$n^{O(1/\epsilon)}$	Metric
MSD outliers	$g^{O(d)}\binom{k+g}{g}\left(\frac{1}{\epsilon}\right)^{O(kd)}$	*or*	*	$(6+\epsilon)$	$n^{O(1/\epsilon)}$	Metric
α-MSR outliers	$g^{O(d)}\binom{k+g}{g}\left(\frac{\alpha}{\epsilon}\right)^{O(kd)}$	$2^{O(d)}n\log n$	*	-		
Fair MSR	$\left(\frac{1}{\epsilon}\right)^{O(kd)}$	$O(kn)+\text{poly}(k)$ *or* $2^{O(d)}n\log n+\text{poly}(k)$	Thm 10	$(3+\epsilon)$	$2^{(k\log k/\epsilon)}n^3$	Metric
Fair MSD	$\left(\frac{1}{\epsilon}\right)^{O(kd)}n$	$2^{O(d)}n\log n+O(kn)$	Thm 12	-		

Table 3: All our results are $(1+\epsilon)$-approximations for metrics with doubling dimension d. The starred items are proved in the full version. The first previous result is due to (Bandyapadhyay, Lochet, and Saurabh 2023). The second and seventh are due to (Chen et al. 2024), who assumed general metric spaces. These previous results were probabilistic, while ours are all deterministic. The third, fifth and sixth results are due to (Buchem et al. 2024). The fourth result is due to (Feldmann and Marx 2018). In (Banerjee et al. 2024) the decomposition results appeared, as well as some additional results which are better under some parameterizations (see Related Work).

the dependence of the exponent to (near) linear in d.

Considering algorithms with run-time exponential in k: A $(2+\epsilon)$-approximation to the metric problem was given in (Chen et al. 2024), and for Euclidean spaces a $(1+\epsilon)$-approximation is known (Bandyapadhyay, Lochet, and Saurabh 2023). The former paper also gave an approximation algorithm for a fair version of MSR.

Turning to α-MSR in generally metrics: A c^{α}-factor algorithm was given in (Charikar and Panigrahy 2001) (for some constant c), and a $(1+\epsilon)$-approximation with run-time $O(kn)+2^{(\frac{\alpha d\log k}{\epsilon})^{O(\min\{\alpha,d\})}}$ was given in (Banerjee et al. 2024). Here too, our results improve upon this bound when $k=d^{o(\min\{\alpha,d\})}$, reducing the dependence of the exponent in d. In (Bandyapadhyay and Varadarajan 2016) a bi-criteria quasi-polynomial time algorithm was given for general metrics.They also show that for large $\alpha\geq\log n$, it is NP-hard to achieve approximation $o(\log n)$.

MSD: In contrast to MSR, MSD is NP-complete even for constant $k\geq 3$ (Brucker 1978). This is true even when the space is restricted to graph metrics. In Euclidean space, an exact algorithm with run-time $n^{O(k)}$ was known (Capoyleas, Rote, and Woeginger 1991), while for metric space only $n^{O(k^2)}$ was previously known (Behsaz and Salavatipour 2015).

Turning to approximation algorithms, the metric MSR algorithm of (Buchem et al. 2024) immediately gives a $(6+\epsilon)$-factor approximation for metric MSD. A 2-approximation is known for constant k (Doddi et al. 2000). The same paper gave a bi-criteria approximation. They also showed that it is NP-hard to obtain a $(2-\epsilon)$-approximation in the metric case for general k. A $(1+\epsilon)$-approximation algorithm with run-time $O(kn)+k^{O(d)}(\log k)^{\tilde{O}(d^2)}2^{(1/\epsilon)^{O(d)}}$ was given in (Banerjee et al. 2024). Our results, when $k=(1/\epsilon)^{o(d)}$, reduce the dependence on d from doubly exponential in d.

2 Preliminaries and Definitions

Definitions and Notation. Throughout, we take (X,d) to denote the input metric space ($n=|X|$), and $d(x,y)$ to denote the pairwise distance metric between $x,y\in X$. The diameter of the point set is denoted $\text{diam}(X)$. The aspect ratio of the space Φ is the ratio between the diameter of the space and the minimum inter-point distance in the space. The distance between a point $y\in Y$ and set $X\subset Y$ is $d(y,X)=\min_{x\in X}d(x,y)$. Let $B(x,r)$ define a ball centered at $x\in X$ of radius $r\geq 0$. We also make use of following notation: For $u\in\mathbb{N}$, $[u]=\{1,\ldots,u\}$. Define the operator: $\lceil\lceil\cdot\rceil\rceil:\mathbb{R}^{\geq 0}\to\mathbb{R}^{\geq 0}$, which rounds $x\in\mathbb{R}^+$ to the smallest power of 2 larger than x: $\lceil\lceil x\rceil\rceil=2^{\lceil\log x\rceil}$. Also set $\lceil\lceil 0\rceil\rceil=0$.

The *doubling dimension* of a metric space X, denoted $d=\text{ddim}(X)$, is the smallest $m>0$ such that every ball of radius r in X can be covered by at most 2^m balls of radius $r/2$. In the context of the Euclidean space, we will use d to denote the dimension of the space, noting that its doubling dimension is $O(d)$.

The optimal solution is denoted OPT and its cost is denoted $\text{cost}(\text{OPT})$.

Nets and Point Hierarchies. An *ϵ-net* of X is a subset $S\subset X$ with the following properties: (i) Packing: S is ϵ-separated, i.e. all distinct $u,v\in S$ satisfy $d(u,v)\geq\epsilon$; and (ii) Covering: every point $x\in X$ is strictly within distance ϵ

of some point $z \in S$, that is $d(x,z) < \epsilon$. A *point hierarchy* (Krauthgamer and Lee 2004) consists of a series of nets S_{2^i} for $i = 0, \ldots, \lceil \log \mathrm{diam}(X) \rceil$, where each S_{2^i} is a 2^i-net of $S_{2^{i-1}}$, and $S_1 = X$. We may refer to S_{2^i} as the i-th level of the hierarchy. We say that two points $x, y \in S_{2^i}$ are c-*neighbors* if $d(x,y) < c \cdot 2^i$. A hierarchy for X which also maintains all c-neighbor pairs (for constant c) can be constructed in time $2^{O(\mathrm{ddim}(X))} n \log n$ (Har-Peled and Mendel 2006; Cole and Gottlieb 2006).

Min-Sum Radii Clustering. Given a metric (X,d) and an integral parameter k, our task is to choose a set of at most k balls $\mathcal{B} = \{B_1, B_2, \ldots B_k\}$, where $B_i = B(x_i, r_i)$, such that their union covers X, i.e. $\cup_{i \in [k]} B_i = X$. The objective to be minimized is the sum of the radii $\sum_{i \in [k]} r_i$. In α-MSR ($\alpha \geq 1$), the cost is $\sum_{i \in [k]} r_i^\alpha$. The above definition requires that center x_i be a point of X; this is the *discrete* version of the MSR problem.

Fair-MSR We study the notion of fairness defined in (Chen et al. 2024). We are given two additional inputs: the first is a coloring, represented by a disjoint partition $Y_1, ..., Y_m$ of X, and the second is a set of integers $k_1, ..., k_m$, such that $\sum_{i=1}^m k_i = k$. The cost function is defined the same way as in MSR, but a solution is feasible if and only if at most k_i of its center points are from Y_i for every $i \in [m]$.

Min-Sum Diameters Clustering. Given a metric (X,d) and an integral parameter k, our task is to segment the points into k disjoint clusters $\mathcal{C} = \{C_1, \ldots, C_k\}$, while minimizing the sum of their diameters: $\sum_{i \in [k]} \mathrm{diam}(C_i)$. In α-MSD ($\alpha \geq 1$), the cost is defined as $\sum_{i \in [k]} (\mathrm{diam}(C_i))^\alpha$.

Mergeable Min-Sum Diameters Clustering. We study the notion of mergeability presented in (Arutyunova and Schmidt 2021) and studied for MSR in (Drexler et al. 2023). We say that a clustering problem is *mergeable* if for any feasible solution, a solution obtained by merging two clusters is still feasible. We assume that there is an efficient procedure which checks the feasibility of a solution in time $f(n)$.

This framework includes many important clustering variants such as clustering with lower bound constraints (Arutyunova and Schmidt 2021). In (Drexler et al. 2023) it was shown that several clustering constraints, and in particular fairness constraints, are mergeable. We refer to these variants as Fair MSD. In particular, let us consider one definition of *balanced* clustering, originally defined in (Chierichetti et al. 2017): Given a partition of X into two colors X_1, X_2 and a parameter $b \in [0,1]$, we say that a cluster C is b-balanced if $\min\{\frac{|C \cap X_2|}{|C \cap X_1|}, \frac{|C \cap X_1|}{|C \cap X_2|}\} \geq b$. This constraint is clearly mergeable and can be validated in time $f(n) = O(n)$.

Clustering with Outliers. A common extension to clustering problems is to allow the solution to include up to g outliers - points in X which are not covered by any cluster. MSR and MSD with outliers are known to have constant factor polynomial approximations with factors $(3+\epsilon)$ and $(6+\epsilon)$ respectively (Buchem et al. 2024), based on the primal-dual rounding approach.

3 Exact Algorithms

Our main contribution in terms of exact algorithms is an $n^{O(k)}$ time algorithm for the MSD problem. For completion of the discussion we begin by recalling that a similar bound trivially holds for MSR.

MSR. For MSR, there is a simple brute-force exact algorithm achieving the following bound:

Proposition 1. *The MSR problem and the α-MSR problem with k clusters can be solved exactly in time $n^{O(k)}$ in general metric spaces. This holds also in the case of g outliers.*

The algorithm enumerates all possible ball centers and radii. As there are $O(n)$ possible centers and $O(n^2)$ possible radii, the bound follows by enumerating all possible solutions of k clusters. For the outliers variant, we consider all solutions which cover at least $n - g$ points

MSD. The case of MSD is considerably more involved. For this problem we adapt and improve upon the algorithm of (Behsaz and Salavatipour 2015). The running time of their proposed algorithm was $n^{O(k^2)}$, and we improve this to $n^{O(k)}$. The key observation of (Behsaz and Salavatipour 2015) is that for all pairs of clusters C_i, C_j in OPT there exists a pair of points $c_j^{(i)} \in C_i, c_i^{(j)} \in C_j$ for which $d(c_j^{(i)}, c_i^{(j)}) > \mathrm{diam}(C_i) + \mathrm{diam}(C_j)$ holds, or else the two clusters may be joined into a single cluster without increasing the cost. We say that $c_j^{(i)}$ is a *witness* for cluster C_i with respect to C_j. The algorithm in (Behsaz and Salavatipour 2015) iterates through all possible combinations of diameters, and for each combination enumerates all possible candidate witness sets for each cluster. For each combination it constructs each cluster as the set of points whose distance from all witnesses is bounded by the chosen diameter bound. The optimal solution is the minimum cost solution wherein every point is assigned to some cluster.

Our improvement over that of (Behsaz and Salavatipour 2015) comes from bounding the number of clusters close to a cluster. We use the following notation and its corresponding property, enabling us to enumerate only a constant number of witnesses per cluster, even in the approximation algorithm:

Definition 1 (Neighborhood)**.** *Let $\mathcal{C}$ be a solution for the MSD problem. The* -neighborhood *of a cluster $C \in \mathcal{C}$, denoted $N^{\mathcal{C}}(C)$, is the set $\{C' | C' \in \mathcal{C} \setminus \{C\}, d(C,C') \leq \mathrm{diam}(C), \mathrm{diam}(C) \leq \mathrm{diam}(C')\}$.*

Lemma 2. *Let $\mathcal{C}$ be a solution for MSD. There is a solution for MSD $\mathcal{C}^\star$ such that $\mathrm{cost}(\mathcal{C}^\star) \leq \mathrm{cost}(\mathcal{C})$ and for every $C \in \mathcal{C}^\star$, $N^{\mathcal{C}^\star}(C) \leq 4$.*

Proof. We initialize $\mathcal{C}^\star$ to be $\mathcal{C}$. If there is $C \in \mathcal{C}^\star$ such that $N^{\mathcal{C}^\star}(C) > 4$, denote by r_1, r_2 the diameters of the two largest clusters in $N^{\mathcal{C}^\star}(C)$. $\mathrm{diam}(C \cup (\bigcup_{C' \in N^{\mathcal{C}^\star}(C))} C') \leq r_1 + r_2 + 3\,\mathrm{diam}(C)$, and $\mathrm{diam}(C) + \sum_{C' \in N^{\mathcal{C}^\star}(C))} \mathrm{diam}(C') \geq r_1 + r_2 + \mathrm{diam}(C) + (|N^{\mathcal{C}^\star}(C))| - 2)\,\mathrm{diam}(C) > r_1 + r_2 + 3\,\mathrm{diam}(C)$, and we may replace the clusters of $N^{\mathcal{C}^\star}(C))$ and C with their union without increasing the cost. Since this process reduces the

number of clusters, it can be done only a finite number of times, after which the condition holds. □

Theorem 3. *The MSD problem with k clusters can be solved exactly in time $n^{O(k)}$ in general metric spaces.*

Proof. Let us begin by presenting the algorithm of (Behsaz and Salavatipour 2015) which computes OPT via brute-force enumeration. Denote by $\mathcal{D}$ the set of all distances between pairs of points in X (including duplicate distances). As OPT may have less than k clusters, we consider every possible number of clusters q between 2 and k, and for each q we iterate through all the possible choices of q values $D_1, ..., D_q$ from $\mathcal{D}$. For every such a choice, we enumerate all possible witness sets, with at most $q-1$ witnesses per cluster. Let $S_1, ..., S_q$ be a candidate set of witnesses: For every S_i we create the set $V_i = \{x \in X | d(x, S_i) \leq D_i\}$, which defines the cluster for this witness set. If $\operatorname{diam}(V_i) > D_i$ we discard the solution as invalid. The algorithm chooses the minimum cost solution from the created solutions covering all points.

We now describe our improved algorithm. As in the original algorithm, we compute OPT via brute force enumeration, but we do so with a few crucial changes. The first change is that we require the candidate diameters $D_1, ..., D_q$, to be non-decreasing: $D_1 \leq ... \leq D_q$. The second change is that we restrict the cardinality of each candidate witness set $S_1, ..., S_q$ to be at most $q^\star$ (and below we will take $q^\star = 4$). The third change is to restrict the choice of witnesses and cluster points: First set $P_1 = X$. Then for every i we choose S_i from P_i only, and construct V_i as before while restricting to P_i: $V_i = \{x \in P_i | d(x, S_i) \leq D_i\}$. We then set $P_{i+1} = P_i \setminus V_i$.

Now for the run-time: There are k choices for q and $O(\binom{n}{2}^k) = O(n^{2k})$ choices for the diameters. At each iteration we choose at most $q^\star$ witnesses per cluster, and so we have $O(\binom{n}{q^\star - 1}^k) = O(n^{k(q^\star)})$ total choices per iteration. The total number of iterations is $O(kn^{2k})$, hence the total run-time is $O(k^2 n^{k(1+q^\star)})$.

We now choose the value of $q^\star$: Let $\mathrm{OPT} = \{C_1, ..., C_q\}$ be an optimal solution to MSD on X with cluster diameters $\operatorname{diam}(C_1) \leq ... \leq \operatorname{diam}(C_q)$ and with $q \leq k$. By applying Lemma 2, we may assume without loss of generaility that for every $C_i \in \mathrm{OPT}$, $|N^{\mathrm{OPT}}(C_i)| \leq 4$, hence we set $q^\star = 4$.

Now, considering OPT as defined above, consider an iteration in which the distances satisfy $D_i = \operatorname{diam}(C_i)$, and S_i is the set of witnesses for C_i with respect to the clusters in $N^{\mathrm{OPT}}(C_i)$ if $N^{\mathrm{OPT}}(C_i) \neq \emptyset$, and an arbitrary point from C_i otherwise.

To complete the proof, we prove by induction that for every $1 \leq i \leq q$ it holds that $V_i = C_i$. Assume by induction that $V_l = C_l$ for every $l < i$, and then we will show that $V_i = C_i$.

First note that since $C_i \cap C_l = \emptyset$ for all $l < i$, we have that $P_i = X \setminus \{C_1, C_2, \ldots C_{i-1}\} \supseteq C_i$. Since $S_i \subseteq C_i$, all points in C_i are within distance D_i from all points in S_i. Since $S_i \neq \emptyset$ and $C_i \subseteq P_i$, we have by the definition of V_i that $C_i \subseteq V_i$. Now, assume by contradiction that $V_i \neq C_i$, and let $u \in V_i \setminus C_i$. Recalling that $V_i \subseteq P_i = X \setminus \{C_1, C_2, \ldots C_{i-1}\}$, it must be that $u \in C_j$ for some $j > i$, implying that $\operatorname{diam}(C_j) \geq \operatorname{diam}(C_i)$. As $S_i \subseteq C_i$ and $u \in C_j$ we have that $d(C_i, C_j) \leq d(S_i, u) \leq D_i = \operatorname{diam}(C_i)$, where the second inequality follows by the definition of V_i, and noting that $u \in V_i$. This means that $C_j \in N^{\mathrm{OPT}}(C_i)$, implying by our assumption on the S_i chosen in the inspected iteration, that S_i contains $c_j^{(i)}$, the witness for C_i with respect to C_j. Its matching witness is $c_i^{(j)} \in C_j$, and from the triangle inequality we obtain $d(c_j^{(i)}, c_i^{(j)}) \leq d(C_i, C_j) + \operatorname{diam}(C_j) \leq \operatorname{diam}(C_i) + \operatorname{diam}(C_j)$, which is a contradiction. We conclude that $V_i = C_i$. □

We also have the following theorem for MSD with outliers. which relies on methods and properties presented in the MSD approximation algorithm (proof is in the full version):

Theorem 4. *The MSD problem with k clusters and g outliers can be solved exactly in time $n^{O(k)}$ in general metric spaces.*

4 Approximation Algorithms

Point Set Decompositions

Our approximation algorithms will require the following decomposition property, first defined in (Banerjee et al. 2024):

Definition 2 (Decomposability). *A problem is ψ-composable in time g if there is an algorithm with run-time g which given X produces a set of components $\mathcal{X}$ of cardinality at most k satisfying:*

1. *Point partition: $\cup_i X_i \in \mathcal{X} = X$ and $X_i \cap X_j = \emptyset$ for all $X_i \neq X_j \in \mathcal{X}$.*
2. *Cluster partition: There exists an optimal solution wherein each cluster C is a subset of some component $X_i \in \mathcal{X}$.*
3. *Component diameter: For all $X_i \in \mathcal{X}$, $\operatorname{diam}(X_i) \leq \psi \cdot \operatorname{cost}(\mathrm{OPT}(X))$.*

If a problem is decomposable, we can create a set of components with favorable properties and treat each component as a separate problem, computing approximate solutions for all values $k' \in [k]$. In (Banerjee et al. 2024), in Theorem 14 and Corollary 69, the following bounds are presented:

Lemma 5. *1. MSR, α-MSR and MSD are $O(k^2)$-decomposable in time $\min\{O(kn), 2^{O(\mathrm{ddim}(X))} n \log n\}$.*
2. MSR, α-MSR and MSD with outliers are $O((k+g)^2)$-decomposable in time $\min\{O((k+g)n), 2^{O(\mathrm{ddim}(X))} n \log n\}$.
3. Fair-MSR is $O(k^2)$-decomposable in time $\min\{2^{O(d)} n \log n, O(kn)\} + \mathrm{poly}(k)$.
4. Mergeable MSD is $O(k)$ decomposable in time $2^{O(d)} n \log n + O(k) f(n)$, where $f(n)$ is the run time of the solution validation process.

Approximation Algorithms – Preliminaries

In this section we describe recursive approximation algorithms which for MSR and MSD with run-time linear or near-linear in n and with additional term, depending on $1/\epsilon$, exponential in k and in d. Our algorithms use Lemma 5 to create a net for X and to obtain bounds on the costs of optimal solutions to both problems. Given the lemma's output to be $X_1, ..., X_{k'}$ with $k' \leq k$, we denote $R = \max_{i \in [k]} \operatorname{diam}(X_i)$. We denote the lower bound on the optimal cost by L, and it is given by $L = \frac{1}{64k^2} \sum_{i=1}^{k} \operatorname{diam}(X_i)$.

We denote the upper bound on the optimal cost by βL, where $\beta = 64k^2$. This notation is used for the algorithm to be flexible in case better bounds of the same nature are found, maybe bounds which require different computation time. We use the following notation:

Definition 3. *With the context of some $T, \epsilon, k > 0$ and a net hierarchy of a component X', let $\epsilon_k^T = \left\lceil \left\lceil \frac{\epsilon T}{k} \right\rceil \right\rceil$ and let $X^{(T)}$ be the ϵ_k^T-net of X' that is: $X^{(T)} = S_{\left\lceil \left\lceil \frac{\epsilon T}{k} \right\rceil \right\rceil}$. For every $x \in X^{(T)}$ we denote by $\tau_T(x)$ the set of points from X' mapped to x in the net $X^{(T)}$.*

For each component X' we create an approximate solution for every $q \in [k]$. We iterate over powers of two which are possible bounds on the cost of the optimal solution. For each such bound, T, we create a set of possible approximations for the ranges of radii/diameters in the solution, starting from $\frac{\epsilon T}{k}$ and going upwards in powers of 2. The recursive [MSR/MSD]Subroutine is used in order to obtain a cover for the net $X^{(T)}$, which is then extended to X'. We denote $T^\star = \lceil \lceil \mathrm{cost}(\mathrm{OPT}) \rceil \rceil$ where OPT is the optimal cost on *the whole space*, and $X^\star = X^{(T^\star)}$. For each problem, we will show that the extension of the solution found on $X^\star$ is a good approximation of the optimal solution on the component.

When all the approximations are computed, an optimal assignment of q values to components can be found in $\mathrm{poly}(k)$ time using a dynamic program: given the optimal solutions for every $q \in [k]$ on two components, for every $q \in [k]$ we find q_1, q_2, such that $q = q_1 + q_2$, and the sum of the costs for the best found solutions using q_1 clusters from one of the components and q_2 clusters from the other components is minimal. We repeat this process iteratively, adding one component at a time. Each step runs in $O(k^2)$ time, and there are at most k steps hence the total time for the process is $O(k^3)$. The cost of the solution in which each component X' is assigned $q = |\,\mathrm{OPT}'\,|$ clusters is a $(1 + O(\epsilon))$ approximation of the optimal cost, hence the framework will produce a good approximation.

The following lemma bounds the number of possible sequences of approximate radii/diameters over which the subroutines iterate (proof in the full version):

Lemma 6. *For every $\delta, \epsilon > 0, k \in \mathbb{N}$, there are $O\left(\frac{\delta^k}{\epsilon^k}\right)$ ways to choose $i_1, ..., i_q$ s.t. $q \leq k$, $i_j \in \mathbb{Z}^{\geq 0}$ for every $1 \leq j \leq q$, with and s.t. $\sum_{j=1}^{q} \frac{\epsilon 2^{i_j}}{k} \leq \delta$.*

Below you will find explanations about the recursive subroutines, and in the paper's full version their pseudo-code is given along formal proofs for their correctness.

Approximation Algorithm for MSR for Bounded k

Our algorithm will first apply the decomposition presented in Lemma 5. Afterwards, for each $q \in [k]$ and for each component in the decomposition, it will run an approximation.

Now we explain the process of approximating MSR on a specific component X' using a specific q. Our algorithm is as follows: we iterate over possible candidates T for $T^\star$. For each guess, we build a solution in a recursive manner: given previously created cover balls, an uncovered point z will be chosen. We iterate over candidate approximate radii for the ball containing the uncovered point. The radii are chosen from the range $\left[\epsilon_k^T, T\right]$. For each candidate radius r', we examine the ball $B(z, 2r')$ in $X^{(T)}$, and iterate over pairs of points (x, y) from it. We create a ball with center x and radius defined according to be $d(x, y)$. We add the ball to the solution and continue to create the cover recursively. When r' is chosen to be larger than the radius of the ball containing z, one of the created balls will be an approximation of the ball containing z. During the run, we keep track of the sum of the approximate radii used so far. If their sum is too high, we stop the recursive process, hence we may use Lemma 6 to bound the possible number of different approximation sequences.

Denote the optimal solution on the component by OPT', and by $\mathrm{OPT}^\star$ the solution on $X^\star$ obtained by moving all the center points of OPT' to center points of $X^\star$, and adding ϵ_k^T to each radius (proof in the full version):

Lemma 7. *When the initial call to MSRSubroutine is performed with $q = |\,\mathrm{OPT}'\,|$ and $T = T^\star$, it produces a solution to MSR on $X^\star$ with $\leq q$ balls and cost $\leq \mathrm{cost}(\mathrm{OPT}^\star)$.*

We have the following observation: If $\mathcal{C}$ be a solution to MSR on $X^{(T)}$, then its extension to X has cost at most $\mathrm{cost}(\mathcal{C}) + |\mathcal{C}|\epsilon_k^T$. This observation implies that the cost of the extension of the solution from Lemma 7 to X' is $\leq \mathrm{cost}(\mathrm{OPT}^\star) + |\,\mathrm{OPT}'\,|\epsilon_k^T \leq \mathrm{cost}(\mathrm{OPT}') + 2|\,\mathrm{OPT}'\,|\epsilon_k^T$. When summing over all components, with each component X' assigned $|\,\mathrm{OPT}'\,|$ clusters, we are guaranteed that the combined cost is $\mathrm{cost}(\mathrm{OPT}) + 2k\epsilon_k^T = (1 + O(\epsilon))\,\mathrm{cost}(\mathrm{OPT})$.

Combining the running time of Lemma 5 and a run time bound proven in the full version we obtain:

Theorem 8. *A $(1+\epsilon)$ approximation of MSR can be obtained in $\min\{O(kn), 2^{O(d)} n \log n\} + \left(\frac{1}{\epsilon}\right)^{O(kd)}$.*

Approximation Algorithm for MSD for Bounded k

The approximation algorithm for MSD combines ideas from the exact algorithm with the framework presented for MSR. As in the MSR approximation, our algorithm will first apply the decomposition presented in Theorem 5. Afterwards, for each $k' \in [k]$ and for each component in the decomposition, it will run an approximation, and the solutions will be merged using a dynamic program as described in 4.

From now on, we refer to the approximation of MSD on a specific component X' using a specific q. Our approximation algorithm is as follows: As in MSR, we iterate over possible bounds on the optimal cost, and for each such cost T we consider the net $X^{(T)}$ as defined above. We aim to obtain an exact solution for MSD on this net: at each call to MSD-Subroutine, we choose a point z for which it is possible that the respective cluster, that is the cluster which contains z in the final approximate solution, wasn't created yet. We iterate through candidate diameters within factor 2 of each other for the diameter of the cluster from the optimal solution containing z. For each candidate diameter r, we create a ball B with radius r around z in $X^{(T)}$, and iterate over all the candidate pairs of points from $X^{(T)}$ which define the cluster's diameter, and candidate choices of $q^\star$ witnesses from this ball, as in

Theorem 3. When trying to create a cluster corresponding to $C \in \mathcal{C}$, we aim to choose witnesses for the clusters of $N^{\mathcal{C}}(C)$. By the choice of the witnesses, the created cluster doesn't intersect larger clusters. While in the exact algorithm it was ensured that smaller clusters were already created and hence the new cluster is exactly the required cluster, here it might not be the case. If the diameter of the created cluster is larger than the chosen candidate diameter we say that the cluster is *enlarged*.

Given a set of created clusters, we perform the following operation iteratively: as long as there is a non-enlarged cluster C created with candidate diameter r, which intersects a cluster C' created with candidate diameter $r' \geq r$, we replace C' with $C' \setminus C$. We call the resulting solution a *refined* solution.

At each stage, the point z is chosen in the following manner: if there are uncovered points, one of them is chosen. Otherwise, if non of the clusters in the refined solution is enlarged we return the refined solution. If there is at least one enlarged cluster in the refined solution, we examine an enlarged cluster with minimal candidate diameter. We choose a pair of points from the cluster which are at maximal distance between each other, and iterate over possible cluster creations with regard to both these points. By the refinement process and by the choice of the witnesses, one of these two points is from a cluster of $\mathcal{C}$ for which a respective cluster wasn't created yet. When the algorithm finishes, we again rely on the refinement process and the choice of witnesses to ensure that the refined solution is the exact solution on $X^{(T)}$.

As in MSR, when processing $X^\star$, with $q = |\operatorname{OPT}'|$, the extension of the solution approximates OPT' within an additive factor of $O(\epsilon_k^{T^\star})$, and the combination of these extensions is an approximate solution on the whole space.

Pseudo-code and analysis of the algorithm is presented in the paper's full version, which along with the run time of Lemma 5 implies:

Theorem 9. *A $(1+\epsilon)$-approximation for MSD can be obtained in $\min\{O(kn), 2^{O(d)} n \log n\} + \left(\frac{1}{\epsilon}\right)^{O(kd)}$ time.*

5 Extension to Other Variants

Fair-MSR

In Fair-MSR as introduced by Chen et al. (Chen et al. 2024), we are given two additional inputs: the first is a disjoint partition $Y_1, ..., Y_m$ of X, and the second is a set of integers $k_1, ..., k_m$, such that $\sum_{i=1}^m k_i = k$. The cost function is the same as in MSR, but a solution is feasible only if at most k_i of its center points are from Y_i for every $i \in [m]$.

From Lemma 5 we know that Fair-MSR is $O(k^2)$ decomposable in time $\min\{O(kn), 2^{O(d)n \log n}\} + \operatorname{poly}(k)n$. We use this decomposition method, and run the same method presented in Section 4, with the some changes. Instead of calculating the approximation for every $q \in [k]$, we call it with every combination of $q_1 \in [k_1], ..., q_m \in [k_m]$, and set the initial q to be $\sum_{i=1}^m q_i$. When merging the solutions obtained on two different components, for each choice of $q_1 \in [k_1], ..., q_m \in [k_m]$ we choose the best solution for each component such that the sum of their respective assignment of centers to each demographic group Y_i amounts to q_i.

Finally, each time we obtain an approximate solution by solving a bipartite matching problem, in a manner inspired by (Chen et al. 2024). The algorithm, along with a proof for the following theorem can be found in the paper's full version:

Theorem 10. *A $(1+\epsilon)$-approximation for Fair MSR can be obtained in $\min\{2^{O(d)} n \log n, \operatorname{poly}(k)n\} + poly(k) + \left(\frac{1}{\epsilon}\right)^{O(kd)}$ time.*

Mergeable MSD

We address the notion of *mergeable* clustering of (Arutyunova and Schmidt 2021). Recall that a clustering problem is *mergeable* if for any feasible solution, a solution obtained by merging two clusters is still feasible. In (Drexler et al. 2023) it was shown that many clustering constraints, including several fairness constrains are mergeable.

We note that all the structural properties we use while solving both exact and approximate MSD rely solely on uniting clusters, and hence they apply for mergeable MSD problems. This implies that our exact algorithm for MSD also works for mergeable MSD problems, with additional run-time for checking the feasibility of each solution. We obtain the following theorem:

Theorem 11. *An exact solution for mergeable MSD can be obtained in time $n^{O(k)} f(n)$ time, where $f(n)$ is the solution validation time. In particular, an exact solution for Fair MSD can be obtained in time $n^{O(k)}$.*

For the approximation algorithm, we have a respective decomposition method, given in Theorem 5. Since we solve our approximation algorithm on each component separately, for the approximation algorithm we consider only mergeable constraints for which the validation process can be applied to each cluster separately. This includes fair MSD.

In the approximation for regular MSD, we find an exact solution on a net of the component, and extend it to the whole component. For this extension to comply with the mergeable constraints when using an ϵ_k^T-net, it is required that the minimal distance between two clusters in the solution is greater than ϵ_k^T. Since our net distance is proportional to the optimal solution's cost, there is an approximate solution which satisfies this condition, and this is the solution we aim to find in the algorithm. We obtain the following theorem (proof in the full version):

Theorem 12. *A $(1+\epsilon)$-approximation for mergeable MSD can be obtained in $\left(\frac{1}{\epsilon}\right)^{O(kd)} f(n) + 2^{O(d)} n \log n + O(k) f(n)$ time. In particular, fair MSD can be solved in time $\left(\frac{1}{\epsilon}\right)^{O(kd)} n + 2^{O(d)} n \log n + O(kn)$*

α-MSR and Clustering With Outliers

The results described in Tables 1 and 3 for α-MSR and for clustering with outliers can be found in the paper's full version.

Acknowledgments

Sandip Banerjee is supported in part by SNSF Grant 200021 200731/1. Yair Bartal and Alon Hovav are supported in part by a grant from the Israeli Science Foundation (2253/22).

References

Arutyunova, A.; and Schmidt, M. 2021. Achieving Anonymity via Weak Lower Bound Constraints for k-Median and k-Means. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Bandyapadhyay, S.; Lochet, W.; and Saurabh, S. 2023. FPT Constant-Approximations for Capacitated Clustering to Minimize the Sum of Cluster Radii. In Chambers, E. W.; and Gudmundsson, J., eds., *39th International Symposium on Computational Geometry (SoCG 2023)*, volume 258 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 12:1–12:14. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-273-0.

Bandyapadhyay, S.; and Varadarajan, K. R. 2016. Approximate Clustering via Metric Partitioning. In Hong, S., ed., *27th International Symposium on Algorithms and Computation, ISAAC 2016, December 12-14, 2016, Sydney, Australia*, volume 64 of *LIPIcs*, 15:1–15:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Banerjee, S.; Bartal, Y.; Gottlieb, L.-A.; and Hovav, A. 2024. Novel Properties of Hierarchical Probabilistic Partitions and Their Algorithmic Applications. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, 1724–1767.

Bartal, Y.; Fandina, O. N.; and Neiman, O. 2022. Covering metric spaces by few trees. *J. Comput. Syst. Sci.*, 130: 26–42.

Behsaz, B.; and Salavatipour, M. 2015. On Minimum sum of radii and diameter clustering. *Algorithmica*, 73(1): 143–165.

Brucker, P. 1978. On the complexity of clustering problems. In *Proc. of the Optimization and Operation research, Lecture notes in Economical and Mathematical Systems*, 45–54.

Buchem, M.; Ettmayr, K.; Rosado, H. K. K.; and Wiese, A. 2024. A $(3+\epsilon)$-approximation algorithm for the minimum sum of radii problem with outliers and extensions for generalized lower bounds. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1738–1765.

Capoyleas, V.; Rote, G.; and Woeginger, G. 1991. Geometric clusterings. *Journal of Algorithms*, 12(2): 341–356.

Charikar, M.; and Panigrahy, R. 2001. Clustering to minimize the sum of cluster diameters. In Vitter, J. S.; Spirakis, P. G.; and Yannakakis, M., eds., *Proceedings on 33rd Annual ACM Symposium on Theory of Computing (STOC 2001), July 6-8, 2001, Heraklion, Crete, Greece*, 1–10. ACM.

Chen, X.; Xu, D.; Xu, Y.; and Zhang, Y. 2024. Parameterized Approximation Algorithms for Sum of Radii Clustering and Variants. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(18): 20666–20673.

Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvitskii, S. 2017. Fair Clustering Through Fairlets. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Cole, R.; and Gottlieb, L.-A. 2006. Searching dynamic point sets in spaces with bounded doubling dimension. In *Proc. of the 38th Ann. ACM Symp. on Theory of Computing (STOC 2006)*, 574–583.

Doddi, S.; Marathe, M.; Taylor, S.; and Widmayer, P. 2000. Approximation algorithms for clustering to minimize the sum of diameters. *Nord. J. Comput*, 7(3): 185–203.

Drexler, L.; Hennes, A.; Lahiri, A.; Schmidt, M.; and Wargalla, J. 2023. Approximating Fair k-Min-Sum-Radii in Euclidean Space. In *WAOA*, 119–133.

Feldmann, A. E.; and Marx, D. 2018. The Parameterized Hardness of the k-Center Problem in Transportation Networks. *CoRR*, abs/1802.08563.

Friggstad, Z.; and Jamshidian, M. 2022. Improved Polynomial-Time Approximations for Clustering with Minimum Sum of Radii or Diameters. In *30th Annual European Symposium on Algorithms (ESA 2022)*, volume 244, 1–14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Gibson, M.; Kanade, G.; Krohn, E.; Pirwani, I.; and Vardarajan, K. 2010. On metric clustering to minimize the sum of radii. *Algorithmica*, 57(3): 484–498.

Hansen, P.; and Jaumard, B. 1987. Minimum sum of diameters clusterings. *Journal of Classification*, 4: 215–226.

Har-Peled, S.; and Mendel, M. 2006. Fast Construction of Nets in Low-Dimensional Metrics and Their Applications. *SIAM Journal on Computing*, 35(5): 1148–1184.

Krauthgamer, R.; and Lee, J. R. 2004. Navigating nets: Simple algorithms for proximity search. In *15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 798–807.

Monma, C.; and Suri, S. 1991. Partitioning points and graphs to minimize the maximum or the sum of diameters. *Graph Theory, Combinatorics and Applications*, 880–912.