Geometric Clustering to Minimize the Sum of Cluster Sizes

Vittorio Bilò[1], Ioannis Caragiannis[2], Christos Kaklamanis[2], and Panagiotis Kanellopoulos[2]

[1] Dipartimento di Matematica "Ennio De Giorgi"
Università di Lecce, Provinciale Lecce-Arnesano, 73100 Lecce, Italy
[2] Research Academic Computer Technology Institute &
Department of Computer Engineering and Informatics,
University of Patras, 26500 Rio, Greece

Abstract. We study geometric versions of the *min-size k-clustering* problem, a clustering problem which generalizes clustering to minimize the sum of cluster radii and has important applications. We prove that the problem can be solved in polynomial time when the points to be clustered are located on a line. For Euclidean spaces of higher dimensions, we show that the problem is NP-hard and present polynomial time approximation schemes. The latter result yields an improved approximation algorithm for the related problem of k-clustering to minimize the sum of cluster diameters.

1 Introduction

Clustering is an area of combinatorial problems which is both algorithmically rich and practically relevant. Several clustering problems have been extensively studied since they have applications in many fields including database systems, image processing, data mining, information retrieval, molecular biology, and more.

Given a set of points X, we call a *cluster* any nonempty subset of X. A set of clusters is a *clustering* for X if each point of X belongs to some cluster. A clustering is called *k-clustering* if it consists of at most k clusters. In general, clustering problems are stated as follows: An instance of such a problem consists of a set X of n points, a distance function dist : $X \times X \rightarrow R$ and an integer k and the objective is to compute a k-clustering of the points in X minimizing $f(C_1, ..., C_k)$, where f is a function defined on the clusters, typically using the distance function dist. Depending on the definition of the function f, many different clustering problems can be defined. The mostly studied ones are the *k-center*, *k-median*, and *k-clustering*. Their objectives are to assign the points to at most k clusters so that the maximum distance from any point to its cluster center (k-center) or the sum of distances from each point to its closest cluster center (k-median) or the sum of all distances between points in the same cluster (k-clustering) is minimized. These problems are NP-hard and several approximation algorithms have been proposed [3,5,13] including polynomial time approximation schemes for geometric instances of these problems [1,2,10,16].

G.S. Brodal and S. Leonardi (Eds.): ESA 2005, LNCS 3669, pp. 460–471, 2005.
© Springer-Verlag Berlin Heidelberg 2005

In this paper, we study a variation of the problem of clustering a set of points into a specific number of clusters so as to minimize the sum of cluster sizes. The size of a cluster may be proportional to the radius/diameter of the cluster, to its area, etc. In particular, minimizing the sum of cluster radii/diameters has been suggested as an alternative to the k-center objective in certain applications so as to avoid the *dissection effect* [8]: using the maximum diameter/radius as the objective sometimes results in objects that should have been placed in the same cluster to be placed in different clusters.

Clustering to minimize the sum of diameters/radii has been studied for points in metric spaces in [6] and [8]. An approximation algorithm which computes a solution with at most $10k$ clusters of cost at most a factor of $O(\log n/k)$ within the optimal solution for k clusters was presented in [8]. This result was improved by Charikar and Panigrahy in [6] where an algorithm that computes a constant approximate solution using at most k clusters is presented. In metric spaces, ρ-approximation algorithms for clustering to minimize the sum of diameters give 2ρ-approximation algorithms for the corresponding radii problem (and vice versa). Negative results include a $2-\epsilon$ inapproximability bound for minimizing the sum of diameters in metric spaces [8] while the complexity of the corresponding radii problem is open. For non-metrics, no approximation bound is possible for diameters in polynomial time unless $P = NP$ even for $k = 3$ [8]. When k is fixed, the optimal solution for radii/diameters can be found in polynomial time by enumerating the $O(n^k)$ possible solutions. The papers [12] and [15] present fast polynomial time algorithms for the case $k = 2$, addressing the Euclidean case as well. Capoyleas et al. [7] study a generalized version of the problem for points on the Euclidean plane and show that, for fixed k and any function of the cluster diameters, it can be solved in polynomial time.

In this paper, we consider geometric versions of the *min-size k-clustering* problem. Formally, an instance (X, F, d, α) of the problem has a set X of n points with rational coordinates on the d-dimensional Euclidean space, a cost function F that associates a fixed non-negative cost with each point, and a constant value α. The objective is to compute a k-clustering $\mathcal{C}$ together with center points $c \in X$ in each cluster C such that $\sum_{C\in\mathcal{C}} COST(C)$ is minimized, where $COST(C)$ is defined as $(\max_{p\in C} \mathrm{dist}(p,c))^\alpha + F_c$ and $\mathrm{dist}(p,c)$ denotes the Euclidean distance between the points p and c. The quantity $\max_{p\in C} \mathrm{dist}(p,c)$ is the *radius* of cluster C with center c.

Besides its importance for clustering optimization, another motivation for studying the min-size k-clustering problem is the following scenario. Assume that a telecommunication agency wishes to give wireless access to users scattered in several locations. This can be achieved by establishing a network of base stations (antennas) to specific locations and setting appropriately the range of each base station such that all the locations are within the range of some station. From the point of view of the agency, establishing a base station incurs a setup cost and an operational cost which is proportional to the range of the station (i.e., the square of the distance of the farthest location within range from the base station). Min-size k-clustering models the problem of minimizing the costs for

building and operating the network. Very recently, we became aware of [14] which studied special cases of min-size k-clustering under this motivation. The authors of [14] study instances $(X, F, d, 1)$ of min-size k-clustering with $k = n$ and fixed costs in $\{0, \infty\}$. They present a dynamic programming algorithm that solves the problem optimally when the points are located on the line and a polynomial-time approximation scheme for points in Euclidean spaces of constant dimensions. This latter result is based on ideas of a dynamic programming algorithm of [9] for approximating the minimum vertex cover of disk graphs.

Min-size k-clustering generalizes the problem of minimizing the sum of radii. We consider the case where k is arbitrary. The result of [6] for metric spaces implies an algorithm with approximation ratio slightly worse than 3^α in our case. We show that the problem is NP-complete in 2-dimensional Euclidean spaces and $\alpha \geq 2$, while a generalized version is solvable in polynomial-time when the points are located on a line. For higher dimensions, we present a polynomial time approximation scheme that computes an $(1 + \epsilon)$-approximate solution using at most k clusters; the running time of our algorithm is $n^{(\alpha/\epsilon)^{O(d)}}$. Our techniques yield a $(2 + \epsilon)$-approximation algorithm for the k-clustering to minimize the sum of cluster diameters. Like [14], our algorithm uses and extends ideas from [9]. Our results are stronger than those in [14] since we assume that k can be arbitrary, that the fixed costs of the points may have arbitrary positive values, and we consider the more general case $\alpha \geq 1$. Our algorithm is guaranteed to find approximate solutions in polynomial time due to structural properties of the optimal or approximate solutions. This is captured by corresponding *Structure Lemmas*.

The rest of the paper is structured as follows. In Section 2 we give complexity results for the problem. We present the algorithm and its analysis in Section 3. Section 4 contains the statements and proofs of the Structure Lemmas. We conclude with some extensions and open problems in Section 5. Due to lack of space, most of the proofs have been omitted.

2 Complexity Results

We first show that the problem is solvable in polynomial time when the points are located on the line.

Theorem 1. *Min-size k-clustering for instances $(X, F, 1, \alpha)$ is in P.*

The proof of this statement follows by expressing the problem as an integer linear program with totally unimodular matrix and concluding that an optimal clustering is obtained by computing a basic solution for the linear program. The statement also holds if the clusters have arbitrary positive costs. Previous results include weaker statements with more complicated proofs [4,14].

In the sequel we consider points in higher dimensions. We can show that two important cases of the problem on the Euclidean plane are NP-hard. The first case is an interesting geometric version of set cover which is also studied in [11]. We have two disjoint sets of points S and T on the Euclidean plane.

We wish to cover all points in T by disks centered in points of S so that the total area of the disks is minimized. It is not difficult to see that this problem is equivalent to the min-size k-clustering with $k = |S \cup T| = n$ and input instance $(S \cup T, F, 2, 2)$ where $F_p = \infty$ if $c \in T$ (this guarantees that points of T should not be cluster centers) and $F_p = 0$ if $c \in S$ (this guarantees that all points of S can be centers of clusters including no points of T). In the instances of the second case that we prove to be NP-hard, all points have zero fixed costs. Our NP-hardness statements follow.

Theorem 2. *Let $(X, F, 2, \alpha)$ be an instance of the problem with $\alpha \geq 2$ and F such that $F_p \in \{0, \infty\}$ for any point $p \in X$. Deciding whether $(X, F, 2, \alpha)$ has any min-size clustering of cost at most K is NP-complete.*

Theorem 3. *Let $(X, F, 2, \alpha)$ be an instance of the problem with $\alpha \geq 2$ and $F_p = 0$ for any point $p \in X$. Deciding whether $(X, F, 2, \alpha)$ has any min-size k-clustering of cost at most K is NP-complete.*

3 The Algorithm

Our algorithm uses the idea of plane subdivision from an algorithm of Erlebach et al. [9] that approximates the minimum vertex cover of disk graphs. Given disks on the plane, the corresponding disk graph is the graph having a node for each disk and an edge between any pair of nodes corresponding to overlapping disks. Although it is not at all related to minimum vertex cover in disk graphs, the min-size k-clustering can be seen as a covering problem with disks as well. We may think of a cluster C with center c as a disk centered at the point c and with radius equal to the maximum distance of c from any point of C (and possibly zero if c is the only point of C). Such a disk has a cost equal to the quantity $(\max_{p \in C} \mathrm{dist}(p, c))^\alpha + F_c$. Now, the min-size k-clustering problem asks for a set of at most k disks with minimum total cost which include (i.e., cover) all points of X.

Before we describe the min-size k-clustering algorithm, we adapt the terminology of [9] to our setting. We use the term cluster instead of the term disk. Fix a positive integer $\lambda > 1$. Consider an instance $(X, F, 2, \alpha)$ of min-size k-clustering and let $\mathcal{D}$ denote the set of all possible n^2 clusters obtained by considering all possible radii for each point in X. Among all clusters of $\mathcal{D}$ with non-zero radius, let r_{min} and r_{max} be the radius of the smallest and the largest cluster, respectively. Partition $\mathcal{D}$ into $L + 1$ levels, where $L = \lfloor \log_{\lambda+1}(r_{max}/r_{min}) \rfloor$. For $0 \leq j \leq L$, level j consists of all clusters d_i having radius r_i such that $(\lambda + 1)^{-j} r_{max} \geq r_i > (\lambda + 1)^{-(j+1)} r_{max}$. Note that the smaller the level, the larger the radii of the clusters are. Thus, the cluster with radius r_{min} will be on level L. We assume that clusters with zero radius belong to level L as well.

For each level j, $0 \leq j \leq L$, impose a grid on the plane consisting of lines that are $2(\lambda+1)^{-j} r_{max}$ apart from each other. The v-th vertical line, for integer v in $(-\infty, \infty)$, is at $x = 2v(\lambda + 1)^{-j} r_{max}$. The h-th horizontal line, for integer h in $(-\infty, \infty)$, is at $y = 2h(\lambda + 1)^{-j} r_{max}$. We say that the v-th vertical line

has index v and that the h-th horizontal line has index h. Furthermore, we say that a cluster d_i with center (x_i, y_i) and radius r_i hits a vertical line at $x = a$ if $a - r_i < x_i \leq a + r_i$. Similarly, we say that d_i hits a horizontal line at $y = b$ if $b - r_i < y_i \leq b + r_i$. Intuitively, by considering clusters as disks, a cluster hits a line if it intersects that line, except if it only touches the line from the left or from below. Note that every cluster can hit at most one horizontal line and at most one vertical line on its level.

Let $0 \leq r, s < \lambda$ and consider the vertical lines whose index modulo λ equals r and the horizontal lines whose index modulo λ equals s. We say that these lines are *active* for (r, s). Consider one particular level j. The lines on level j that are active for (r, s) partition the plane into squares. More precisely, for consecutive active vertical lines at $x = a_1$ and $x = a_2$ and consecutive active horizontal lines at $y = b_1$ and $y = b_2$, one square $\{(x, y) | a_1 < x \leq a_2, b_1 < y \leq b_2\}$ is obtained. We refer to these squares on level j as *j-squares*. As observed in [9], for any j, $0 \leq j < L$, every $(j + 1)$-square is completely contained in some j-square. An example is depicted in Figure 1.

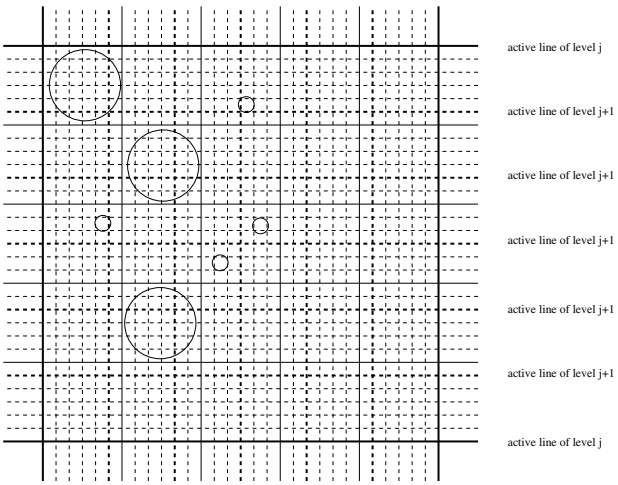

Fig. 1. An example of the plane subdivision for $\lambda = 5$. The disks shown represent clusters of level j of the minimum and maximum possible radius.

A j-square S is relevant if there exists at least one cluster of level j in $\mathcal{D}$ containing a point $p \in S \cap X$. Observe that the number of relevant squares is polynomial in n, since the number of clusters is n^2 and a cluster may cover points in at most 4 squares of its level. For a relevant j-square S and a relevant j'-square S' with $j' > j$, we say that S' is a *child square* of S (and S is a *parent* of S') if S' is contained in S and there is no relevant j''-square S'' with $j' > j'' > j$, such that S' is contained in S'' and S'' is contained in S. It can be easily seen that the number of relevant 0-squares is at most 4; these are the only squares

without a parent. We show the following property which holds specifically for instances of min-size k-clustering.

Lemma 1. *Each relevant square has at most $O(\lambda^4)$ child squares.*

Proof. Clearly, a square S of level j and of side length ℓ may have at most $(\lambda+1)^4$ child squares of levels $j+1$ and $j+2$. If S has more than $(\lambda+1)^4$ child squares, then it should have child squares of level at least $j+3$. We will show that the number of child squares of S of level at least $j+3$ is at most $\frac{16}{\pi}(2(\lambda+1)^2+1)^2$.

Pick a square S' of smallest level $j' \geq j+3$ among the child squares of S and let p be a point contained in it. Then, all other points will be at distance either smaller than $\frac{\ell}{2(\lambda+1)^{j'-j+1}}$ or at least $\frac{\ell}{2(\lambda+1)^2}$, otherwise the j''-square S'' containing S' with $j < j'' < j'$ would be relevant and, hence, S'', instead of S', would be a child of S. Now, observe that, within a disk of radius $\frac{\ell}{4(\lambda+1)^2}$ centered at p, there can be at most four child squares of S of level at least $j+3$, including S'; this is the maximum number of squares that may have one point at distance smaller than $\frac{\ell}{2(\lambda+1)^{j'-j+1}}$ from p. Repeat recursively this procedure for the child squares of S of level at least $j+3$ which are not contained in the disk until all squares of level at least $j+3$ have been included in disks. The disks do not overlap, otherwise this would mean that the center of some disk which is a point in a square of level $j+3$ has distance smaller than $\frac{\ell}{2(\lambda+1)^2}$ and at least $\frac{\ell}{2(\lambda+1)^{j''-j+1}}$ from some other point. Also, they all have their centers in S, thus they are all contained in the square of side length $\left(1+\frac{1}{2(\lambda+1)^2}\right)\ell$. Hence, their number is at most

$$\frac{\left(1+\frac{1}{2(\lambda+1)^2}\right)^2 \ell^2}{\pi\left(\frac{\ell}{4(\lambda+1)^2}\right)^2} \leq \frac{4}{\pi}(2(\lambda+1)^2+1)^2,$$

and the number of child squares of S of level at least $j+3$ cannot exceed $\frac{16}{\pi}(2(\lambda+1)^2+1)^2$. □

Consider some j-square S and denote by I^S the set of clusters in $\mathcal{D}$ intersecting S. We denote by $I^S_{<j}$ the set of clusters in I^S having level smaller than j and define $I^S_{\leq j}$, $I^S_{=j}$, $I^S_{\geq j}$ and $I^S_{>j}$ analogously. We say that a set $C \subseteq I^S$ is a *pseudoclustering* of S if for any point $p \in X \cap S$ there exists a cluster in C containing p. For any pseudoclustering C of S, call $I^S_{<j} \cap C$ the *projection* of C onto $I^S_{<j}$ (and similarly for $I^S_{\leq j}$).

Now, we are ready to describe the algorithm. Given an instance $(X, F, 2, \alpha)$ of min-size k-clustering, the algorithm assigns levels to all possible clusters defined by X and implicitly defines horizontal and vertical lines on the plane as discussed above. Then, for each possible value of $r, s \in \{0, ..., \lambda-1\}$, it executes an iteration. In each iteration, a k-clustering is computed; the best k-clustering among all iterations is output as the final solution. In each iteration associated with r, s, the algorithm processes all relevant squares defined by the plane subdivision according to r and s in a bottom-up fashion (i.e., in decreasing order of levels). At

a relevant j-square S, the projections of polynomially many pseudoclusterings of S are enumerated. During this enumeration process, a table $Table_S$ is constructed by looking up the respective entries stored in tables at children of S. The entry $Table_S(P, i)$ for a projection $P \subseteq I^S_{<j}$ of a pseudoclustering of S onto $I^S_{<j}$ and an integer i such that $1 \leq i \leq k$, will be a set $J \subseteq I^S_{\geq j}$ such that $P \cup J$ is a pseudoclustering of S with exactly i clusters. At the end of each iteration, the algorithm computes a k-clustering by enumerating all clusterings obtained by choosing entries from each table $Table_S$ taken over all relevant squares S having no parent.

```
Table_S ← ∅
I^S_≤j ← all clusters in D of level at most j intersecting S
for all Q ⊆ I^S_≤j such that |Q| ≤ min{ξ, k} do
    J ← {D ∈ Q | D has level j}
    P ← {D ∈ Q | D has level smaller than j}
    if S has no children then
        Table_S(P, |Q|) ← J
    else
        let S_1, S_2, ..., S_t be the child squares of S
        for each child square S_y do
            P'(S_y) ← {D ∈ Q | D intersects S_y}
        for each possible combination of (i_1, i_2, ..., i_t)
        with 1 ≤ i_y ≤ k for y = 1, ..., t do
            J' ← J ∪ ⋃_{y=1}^{t} Table_{S_y}(P'(S_y), i_y)
            i' = |P ∪ J'|
            if i' ≤ k and P ∪ J' is a pseudoclustering of S then
                if Table_S(P, i') is undefined
                or ω(J') < ω(Table_S(P, i')) then
                    Table_S(P, i') ← J'
```

Fig. 2. The pseudocode for computing $Table_S$ once the tables $Table_{S'}$ have been computed for all children S' of S and all values of i

In Figure 2, we present the pseudocode for computing $Table_S$ once the tables $Table_{S'}$ have been computed for all children S' of S and all values of i. The parameter ξ is used to constrain the size of pseudoclusterings of S considered. We use $\omega(.)$ to denote the cost of a cluster or the total cost of a set of clusters.

The algorithm executes λ^2 iterations. In each iteration, at most $O(n^2)$ relevant squares are processed. Using Lemma 1, we can easily see that the time required for computing the table entries for each relevant square is at most $n^{O(\lambda^4+\xi)}$. Since the number of relevant squares having no parent in each iteration is at most 4, the last step of each iteration completes in polynomial time. Overall, the running time of the algorithm is $n^{O(\lambda^4+\xi)}$.

In the following, we present the main arguments for analyzing the performance of the algorithm. Let $(X, F, 2, \alpha)$ be an instance of min-size k-clustering and consider all solutions which, for any square of side ℓ contain at most ξ clusters of radius at least $\frac{\ell}{2(\lambda+1)^2}$ that can include all the points of X in the square.

We call such solutions ξ-solutions for instance $(X, F, 2, \alpha)$. Clearly, for any relevant j-square defined by the plane subdivision according to r, s, a ξ-solution for $(X, F, 2, \alpha)$ contains at most ξ clusters of level at most j covering all the points of X in the square. The proof of the efficiency of the algorithm will be based on the comparison of the cost of the solutions obtained with the cost of the best ξ-solution. This will follow by Lemmas 2 and 3. First, in Lemma 2, we show that the cost of the solution computed by the algorithm in an iteration associated with r, s is upper-bounded by a quantity defined as a function of the cost of the clusters in the best ξ-solution and the plane subdivision defined by r, s. Then, in Lemma 3, we show that there are values of r, s such that this latter quantity (and, consequently, the cost of the best solution computed by the algorithm) is not much larger than the cost of the best ξ-solution.

Denote by C^* the best ξ-solution of instance $(X, F, 2, \alpha)$. For any relevant j-square S, denote by $C^*(S)$ the clusters of level j in C^* intersecting S.

Lemma 2. *Let $r, s \in \{0, ..., \lambda - 1\}$ and $\mathcal{S}(r, s)$ be the set of relevant squares defined by r, s and X. In the iteration associated with r, s, the algorithm computes a k-clustering $A(r, s)$ of X of cost $\omega(A(r, s)) \leq \sum_{S \in \mathcal{S}(r,s)} \omega(C^*(S))$.*

Proof. Since C^* is a ξ-solution, we may assign each point to exactly one cluster so that all points are assigned to some cluster and the number of clusters intersecting with some square S which have been assigned points contained in S is at most ξ. We call a cluster intersecting with a relevant square S and having been assigned a point of S, a cluster associated with S.

For any relevant j-square S, let C^S be the set of clusters in C^* associated with S. Define $C^S_{<j}$, $C^S_{\leq j}$ and $C^S_{=j}$ as usual. We claim that after $Table_S$ has been computed, it holds

$$\omega(Table_S(C^S_{<j}, |C^S|)) \leq \sum_{S' \prec S} \omega(C^*(S')), \tag{1}$$

where $S' \prec S$ denotes that S' is a relevant square that is contained in S. Note that $S \prec S$.

The proof is by induction on the order in which the relevant squares are processed during an iteration. It is trivially true when S has no children. Assume that the algorithm is about to process the relevant j-square S and that (1) holds for all squares processed before S. In one of the iterations of the outer loop in the pseudocode of Figure 2, we have $Q = C^S_{\leq j}$ (and $J = C^S_{=j}$). In this iteration, consider the combination $(i_1, i_2, ..., i_t)$ such that $P'(S_y) = C^{S_y}_{\leq j}$ and $i_y = |C^{S_y}|$ for any $1 \leq y \leq t$. Observe that for each j'-square which is a child of S, it is $C^{S'}_{\leq j} = C^{S'}_{<j'}$. Also, clearly, it is $C^S_{=j} \subseteq C^*(S)$. Thus, the minimum cost set J' such that $P \cup J'$ is a pseudoclustering of S and $|P \cup J'| = |C^S|$ assigned to the entry $Table_S(C^S_{<j}, |C^S|)$ has cost at most

$$\sum_{S' \text{ child of } S} \omega(Table_{S'}(C^{S'}_{\leq j}, |C^{S'}|)) + \omega(C^*(S)) \leq \sum_{S' \prec S} \omega(C^*(S')),$$

and, hence, (1) holds also for S.

Finally, let $\mathcal{S}_0(r,s)$ be the set of all relevant squares without a parent. Once again the algorithm performs a complete enumeration of all possible solutions obtained by choosing exactly one entry from each table $Table_S$ for all $S \in \mathcal{S}_0(r,s)$. By applying the same argument used above and using the fact that for any relevant j-square $S \in \mathcal{S}_0(r,s)$ it is $C^S_{<j} = \emptyset$, we obtain that $\omega(A(r,s)) \leq \sum_{S\in\mathcal{S}_0(r,s)} \omega(Table_S(\emptyset, |C^S|)) \leq \sum_{S\in\mathcal{S}(r,s)} \omega(C^*(S))$. □

Lemma 3. *There exist $r, s \in \{0, 1, ..., \lambda - 1\}$ such that $\sum_{S\in\mathcal{S}(r,s)} \omega(C^*(S)) \leq \left(1 + \frac{6}{\lambda}\right) \omega(C^*)$.*

Similar statements with Lemma 3 are proved in [9,14]. So far (by combining Lemmas 2 and 3), we have bounded the cost of the best solution computed by the algorithm after all iterations in terms of the cost of the best ξ-solution. In the next section, we prove that for any instance $(X, F, 2, \alpha)$ the optimal solution (for $\alpha = 1$) or approximate solutions (for $\alpha > 1$) are essentially ξ-solutions. Combining the analysis above with Lemmas 4 and 5, we can bound the cost of the solution computed by our algorithm in terms of the cost of the optimal solution. By appropriately setting the parameters λ and ξ in terms of ϵ (for any $\epsilon > 0$), we obtain the following theorems.

Theorem 4. *There exists an algorithm for min-size k-clustering which, for each instance $(X, F, 2, 1)$ of the problem, computes an $(1+\epsilon)$-approximate solution in time $n^{O(1/\epsilon^4)}$ for any $\epsilon > 0$.*

Theorem 5. *There exists an algorithm for min-size k-clustering which, for each instance $(X, F, 2, \alpha)$ of the problem, computes an $(1+\epsilon)$-approximate solution in time $n^{O(\alpha^4/\epsilon^6)}$ for any $\epsilon > 0$.*

4 The Structure Lemmas

The following lemmas imply that for any instance $(X, F, 2, \alpha)$ of the min-size k-clustering problem, there exist constant values for ξ such that any optimal solution (for $\alpha = 1$) or at least a particular approximate solution (for $\alpha > 1$) are essentially ξ-solutions (and, hence, the best ξ-solution is optimal or almost optimal, respectively).

Lemma 4 (Structure Lemma). *For any integer constant $\lambda > 1$, there exists a constant $\xi = \xi(\lambda) = O(\lambda^4)$ such that the following is true: For any square S of side length ℓ, any optimal solution for any instance $(X, F, 2, 1)$ of the min-size k-clustering problem, contains at most ξ clusters of radius at least $\frac{\ell}{2(\lambda+1)^2}$ which intersect with S.*

A slightly different version of this Structure Lemma can also be found in [14]. For the case $\alpha > 1$, we cannot show a statement as strong as Lemma 4. Actually, it can be shown that there exist instances $(X, F, 2, \alpha)$ with $\alpha > 1$ and squares S of side length ℓ such that optimal solutions for $(X, F, 2, \alpha)$ contain

an unbounded number of clusters of radius at least $\frac{\ell}{2(\lambda+1)^2}$ intersecting with S. However, we can prove the next *Approximate Structure Lemma* which states that an approximate solution is a ξ-solution and, hence, it suffices for our purposes.

Lemma 5 (Approximate Structure Lemma). *For any constants $\gamma > 0$, $\alpha > 1$, and integer $\lambda > 1$, there exists a constant $\xi = \xi(\lambda, \alpha, \gamma) = O\left(\frac{\alpha^2\lambda^4}{\gamma^2}\right)$ such that the following is true: Any instance $(X, F, 2, \alpha)$ of the min-size k-clustering problem has an $(1+\gamma)^\alpha$-approximate solution which, for any square S of side ℓ, contains a subset of at most ξ clusters of radius at least $\frac{\ell}{2(\lambda+1)^2}$ which contain all points in S.*

Proof. Consider an instance $(X, F, 2, \alpha)$ of the min-size k-clustering problem and an optimal solution D^*_{OPT} for $(X, F, 2, \alpha)$. Let ψ_{OPT} be a function that assigns to each point of X a cluster of D^*_{OPT} containing this point. We obtain a $(1+\gamma)^\alpha$-approximate solution D^* by increasing the radius of each disk in D^*_{OPT} by a factor of $1+\gamma$. Define the assignment ψ which assigns each point of X to the smallest cluster (i.e., the one with the smallest radius) of D^* that contains it. We will show that, for any square of side length ℓ, the number of clusters of D^* of radius at least $\frac{\ell}{2(\lambda+1)^2}$ which are assigned by ψ to points of S is at most

$$\xi(\lambda, \alpha, \gamma) = \left(\left(6\sqrt{2} + \frac{4\sqrt{2}}{\gamma}\right)\frac{\alpha(1+\gamma)(\lambda+1)^2}{\ln 2} + 1\right)^2 + 1.$$

Let S be a square of side length ℓ. Denote by X_1 and X_2 the sets of points of S assigned by ψ_{OPT} to clusters of D^*_{OPT} of radii smaller than $\ell\sqrt{2}/\gamma$ and at least $\ell\sqrt{2}/\gamma$, respectively. All points in X_1 are assigned to clusters of D^* of radii smaller than $\ell\sqrt{2}(1+1/\gamma)$ by ψ. Furthermore, the radius of the clusters of D^*_{OPT} to which points of X_2 are assigned by ψ_{OPT} is increased by at least $\ell\sqrt{2}$ and, hence, the resulting clusters of D^* cover the whole square. Among these clusters, denote by d the one with the smallest radius. The points of X_2 (if any) will be assigned either to cluster d or to clusters of radius smaller than $\ell\sqrt{2}(1+1/\gamma)$.

Now assume that more than $\xi(\lambda, \alpha, \gamma)$ clusters of D^* are assigned to points of the square S by ψ. This means that more than $\xi(\lambda, \alpha, \gamma) - 1$ clusters of radius larger than $\frac{\ell}{2(\lambda+1)^2}$ and at most $\ell\sqrt{2}/\gamma$ have their centers at distance at most $\left(\frac{3}{\sqrt{2}} + \frac{\sqrt{2}}{\gamma}\right)\ell$ from the center O of the square, otherwise, these clusters would not cover any point of S. Now shrink all these clusters around their centers to obtain disks of radius $\frac{2^{1/\alpha}-1}{4(1+\gamma)(\lambda+1)^2}\ell$. Let D' be the set of shrunk disks. We claim that any two disks of D' are disjoint. Assume otherwise and consider two disks of D' centered at points c_1 and c_2 of distance δ smaller than $\frac{(2^{1/\alpha}-1)\ell}{2(1+\gamma)(\lambda+1)^2}$. Let d_1 and d_2 be the clusters centered at c_1 and c_2 in the optimal solution D^*_{OPT} and let r and R be their radii. Without loss of generality, assume that $r \leq R$. Clearly, $r, R \geq \frac{\ell}{2(\lambda+1)^2(1+\gamma)}$. If $R \geq r + \delta$, then this means that the cluster d_2 could include all points included in the cluster d_1, hence the solution D^* would not be optimal. If $R < r + \delta$, then we can include all points included in clusters

d_1 and d_2 in the solution D^*_{OPT} by increasing the radius of the cluster d_2 to $r+\delta$ and removing cluster d_1 from D^*_{OPT}. The new cost of cluster d_2 is now

$$F_{c_2} + (r+\delta)^\alpha < F_{c_2} + \left(r + \frac{2^{1/\alpha}-1}{2(1+\gamma)(\lambda+1)^2}\ell\right)^\alpha \leq F_{c_2} + (r + (2^{1/\alpha}-1)r)^\alpha$$
$$\leq F_{c_2} + 2r^\alpha \leq F_{c_1} + r^\alpha + F_{c_2} + R^\alpha$$

which means that D^* is not optimal. Hence, all disks of D' are disjoint. By their definition, they are contained in a disk d' with radius $\left(\frac{3}{\sqrt{2}} + \frac{\sqrt{2}}{\gamma} + \frac{2^{1/\alpha}-1}{4(1+\gamma)(\lambda+1)^2}\right)\ell$ centered at O. Since they are disjoint, their total area is more than

$$(\xi(\lambda,\alpha,\gamma)-1)\pi \left(\frac{(2^{1/\alpha}-1)\ell}{4(1+\gamma)(\lambda+1)^2}\right)^2$$
$$\geq \left(\left(6\sqrt{2} + \frac{4\sqrt{2}}{\gamma}\right)\frac{\alpha(1+\gamma)(\lambda+1)^2}{\ln 2} + 1\right)^2 \pi \left(\frac{(2^{1/\alpha}-1)\ell}{4(1+\gamma)(\lambda+1)^2}\right)^2$$
$$\geq \left(\left(6\sqrt{2} + \frac{4\sqrt{2}}{\gamma}\right)\frac{(1+\gamma)(\lambda+1)^2}{2^{1/\alpha}-1} + 1\right)^2 \pi \left(\frac{(2^{1/\alpha}-1)\ell}{4(1+\gamma)(\lambda+1)^2}\right)^2$$
$$\geq \pi \left(\frac{3}{\sqrt{2}} + \frac{\sqrt{2}}{\gamma} + \frac{2^{1/\alpha}-1}{4(1+\gamma))(\lambda+1)^4}\right)^2 \ell^2$$

which contradicts the fact that they are completely contained in the disk d'. Hence, the number of clusters of D^* of radius at least $\frac{\ell}{2(\lambda+1)^2}$ which are assigned to points of S cannot exceed $\xi(\lambda,\alpha,\gamma)$. □

5 Extensions and Open Problems

Our techniques naturally extend to higher dimensions by using similar subdivisions of Euclidean spaces. Again, appropriate Structure Lemmas can be shown with slightly more complicated arguments. We can show the following statement.

Theorem 6. *There exists an algorithm for min-size k-clustering which, for each instance (X, F, d, α) of the problem, computes an $(1+\epsilon)$-approximate solution in time $n^{(\alpha/\epsilon)^{O(d)}}$ for any $\epsilon > 0$, $\alpha \geq 1$, and constant integer $d \geq 2$.*

The most important open problem is to explore the complexity of the problems in the case $\alpha = 1$. problems are still open for metric spaces as well; the best known approximability result is the constant approximation algorithm of [6].

In k-clustering to minimize the sum of cluster diameters, the cluster centers need not necessarily be points of X. Our polynomial-time approximation scheme for min-size k-clustering with $\alpha = 1$ yield a $(2+\epsilon)$-approximation algorithm for any constant dimension. To our knowledge, this is the best approximation guarantee for arbitrary k. Further improvements are also possible. Again, the complexity of the problem in multidimensional Euclidean spaces is still open.

References

1. S. Arora, P. Raghavan, and S. Rao. Approximation schemes for the Euclidean k-medians and related problems. In *Proc. of the 30th ACM Symposium on Theory of Computing (STOC '98)*, pp. 106-113, 1998.
2. M. Bădoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. In *Proc. of the 34th Annual ACM Symposium on Theory of Computing (STOC '02)*, pp. 250-257, 2002.
3. Y. Bartal, M. Charikar, and D. Raz. Approximating min-sum k-clustering in metric spaces. In *Proc. of the 33rd Annual ACM Symposium on Theory of computing (STOC '01)*, p.11-20, 2001.
4. P. Brucker. On the complexity of clustering problems. *Optimization and Operations Research*, Lecture Notes in Economics and Mathematical Sciences, Vol. 157, pp. 45-54, 1978.
5. M. Charikar, S. Guha, E. Tardos, and D. S. Shmoys. A constant factor approximation algorithm for the k-median problem. *Journal of Computer and Systems Sciences*, Vol. 65 (1), pp. 129-149, 2002.
6. M. Charikar and R. Panigrahy. Clustering to minimize the sum of cluster diameters. *Journal of Computer and Systems Sciences*, Vol. 68 (2), pp. 417-441, 2004.
7. V. Capoyleas, G. Rote, and G. J. Woeginger. Geometric Clusterings. *Journal of Algorithms*, Vol. 12(2), pp. 341-356, 1991.
8. S. R. Doddi, M. V. Marathe, S. S. Ravi, D. S. Taylor, and P. Widmayer. Approximation algorithms for clustering to minimize the sum of diameters. *Nordic Journal of Computing*, Vol. 7(3), pp. 185-203 , 2000.
9. T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation schemes for geometric graphs. In *Proc of the 12th Annual Symposium on Discrete Algorithms (SODA '01)*, pp. 671-679, 2001.
10. W. Fernandez de la Vega, M. Karpinski, C. Kenyon, and Y. Rabani. Approximation schemes for clustering problems. In *Proc. of the 35th Annual ACM Symposium on Theory of Computing (STOC '03)*, pp. 50-58, 2003.
11. A. Freund and D. Rawitz. Combinatorial interpretations of dual fitting and primal fitting. In *Proc. of the First International Workshop on Approximation and Online Algorithms (WAOA '03)*, LNCS 2909, Springer, pp. 137-150, 2003.
12. P. Hansen and B. Jaumard. Minimum sum of diameters clustering. *Journal of Classification*, Vol. 4, pp. 215-226, 1987.
13. K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual scheme and Lagrangian relaxation. *Journal of the ACM*, Vol. 48, pp. 274-296, 2001.
14. N. Lev-Tov and D. Peleg. Polynomial time approximation schemes for base station coverage with minimum total radii. *Computer Networks*, Vol. 47, pp. 489-501, 2005.
15. C. L. Monma and S. Suri. Partitioning points and graphs to minimize the maximum or the sum of diameters. *Graph Theory, Combinatorics and Applications*, John Wiley and Sons, pp. 880-912, 1991.
16. R. Ostrovsky and Y. Rabani. Polynomial-time approximation schemes for geometric clustering problems. *Journal of the ACM*, Vol. 49(2), pp. 139-156, 2002.