

Analysis of Agglomerative Clustering

Marcel R. Ackermann · Johannes Blömer ·
Daniel Kuntze · Christian Sohler

Received: 2 April 2012 / Accepted: 9 November 2012 / Published online: 12 December 2012
© Springer Science+Business Media New York 2012

Abstract The diameter k -clustering problem is the problem of partitioning a finite subset of \mathbb{R}^d into k subsets called clusters such that the maximum diameter of the clusters is minimized. One early clustering algorithm that computes a hierarchy of approximate solutions to this problem (for all values of k) is the agglomerative clustering algorithm with the complete linkage strategy. For decades, this algorithm has been widely used by practitioners. However, it is not well studied theoretically. In this paper, we analyze the agglomerative complete linkage clustering algorithm. Assuming that the dimension d is a constant, we show that for any k the solution computed by this algorithm is an $O(\log k)$ -approximation to the diameter k -clustering problem. Our analysis does not only hold for the Euclidean distance but for any metric that is based on a norm. Furthermore, we analyze the closely related k -center and dis-

A preliminary version of this article appeared in *Proceedings of the 28th International Symposium on Theoretical Aspects of Computer Science (STACS'11)*, March 2011, pp. 308–319.

Work was done while M.R. Ackermann was at Department of Computer Science, University of Paderborn, Germany.

For all four authors this research was supported by the German Research Foundation (DFG), grants BL 314/6-2 and SO 514/4-2.

M.R. Ackermann

Schloss Dagstuhl—Leibniz Center for Informatics, 66687 Wadern, Germany
e-mail: mra@dbis.uni-trier.de

J. Blömer · D. Kuntze (✉)

Department of Computer Science, University of Paderborn, 33098 Paderborn, Germany
e-mail: kuntze@uni-paderborn.de

J. Blömer

e-mail: bloemer@uni-paderborn.de

C. Sohler

Department of Computer Science, TU Dortmund, 44221 Dortmund, Germany
e-mail: christian.sohler@tu-dortmund.de

crete k -center problem. For the corresponding agglomerative algorithms, we deduce an approximation factor of $O(\log k)$ as well.

Keywords Agglomerative clustering · Hierarchical clustering · Complete linkage · Approximation guarantees

1 Introduction

Clustering is the process of partitioning a set of objects into subsets (called clusters) such that each subset contains similar objects and objects in different subsets are dissimilar. There are many applications for clustering, including data compression [14], analysis of gene expression data [5], anomaly detection [11], and structuring results of search engines [2]. For every application, a proper objective function is used to measure the quality of a clustering. One particular objective function is the largest diameter of the clusters. If the desired number of clusters k is given, we call the problem of minimizing this objective function the *diameter k -clustering problem*.

One of the earliest and most widely used clustering strategies is agglomerative clustering. The history of agglomerative clustering goes back at least to the 1950s (see for example [7, 12]). Later, biological taxonomy became one of the driving forces of cluster analysis. In [15] the authors, who where the first biologists using computers to classify organisms, discuss several agglomerative clustering methods.

Agglomerative clustering is a bottom-up clustering process. At the beginning, every input object forms its own cluster. In each subsequent step, the two ‘closest’ clusters will be merged until only one cluster remains. This clustering process creates a hierarchy of clusters, such that for any two different clusters A and B from possibly different levels of the hierarchy we either have $A \cap B = \emptyset$, $A \subset B$, or $B \subset A$. Such a hierarchy is useful in many applications, for example, when one is interested in hereditary properties of the clusters (as in some bioinformatics applications) or if the exact number of clusters is a priori unknown.

In order to define the agglomerative strategy properly, we have to specify a distance measure between clusters. Given a distance function between data objects, the following distance measures between clusters are frequently used. In the *single linkage strategy*, the distance between two clusters is defined as the distance between their closest pair of data objects. Using this strategy is equivalent to computing a minimum spanning tree of the graph induced by the distance function using Kruskal’s algorithm. In case of the *complete linkage strategy*, the distance between two clusters is defined as the distance between their farthest pair of data objects. In the *average linkage strategy* the distance is defined as the average distance between data objects from the two clusters.

1.1 Related Work

In this paper, we study the agglomerative clustering algorithm using the complete linkage strategy to find a hierarchical clustering of n points from \mathbb{R}^d . The running time is obviously polynomial in the description length of the input. Therefore,

our only goal in this paper is to give an approximation guarantee for the diameter k -clustering problem. The approximation guarantee is given by a factor α such that the cost of the k -clustering computed by the algorithm is at most α times the cost of an optimal k -clustering. Although the agglomerative complete linkage clustering algorithm is widely used, there are only few theoretical results considering the quality of the clustering computed by this algorithm. It is known that there exists a certain metric distance function such that this algorithm computes a k -clustering with an approximation factor of $\Omega(\log k)$ [4]. However, prior to the analysis we present in this paper, no non-trivial upper bound for the approximation guarantee of the classical complete linkage agglomerative clustering algorithm was known, and deriving such a bound has been discussed as one of the open problems in [4].

The diameter k -clustering problem is closely related to the k -center problem. In this problem, we are searching for k centers and the objective is to minimize the maximum distance of any input point to the nearest center. When the centers are restricted to come from the set of the input points, the problem is called the *discrete k -center problem*. It is known that for metric distance functions the costs of optimal solutions to all three problems are within a factor of 2 from each other.

For the Euclidean case, we know that for fixed k , i.e. when we are not interested in a hierarchy of clusterings, the diameter k -clustering problem and the k -center problem are \mathcal{NP} -hard. In fact, it is already \mathcal{NP} -hard to approximate both problems with an approximation factor below 1.96 and 1.82 respectively [6].

Furthermore, there exist provably good approximation algorithms in this case. For the discrete k -center problem, a simple 2-approximation algorithm is known for metric spaces [9], which immediately yields a 4-approximation algorithm for the diameter k -clustering problem. For the k -center problem, a variety of results is known. For example, for the Euclidean metric in [1] a $(1 + \epsilon)$ -approximation algorithm with running time $2^{O(\frac{k \log k}{\epsilon^2})} dn$ is shown. This implies a $(2 + \epsilon)$ -approximation algorithm with the same running time for the diameter k -clustering problem.

Also, for metric spaces a hierarchical clustering strategy with an approximation guarantee of 8 for the discrete k -center problem is known [4]. This implies an algorithm with an approximation guarantee of 16 for the diameter k -clustering problem.

This paper as well as all of the above mentioned work is about static clustering, i.e. in the problem definition we are given the whole set of input points at once. An alternative model of the input data is to consider sequences of points that are given one after another. In [3], the authors discuss clustering in a so-called *incremental clustering* model. They give an algorithm with constant approximation factor that maintains a hierarchical clustering while new points are added to the input set. Furthermore, they show a lower bound of $\Omega(\log k)$ for the agglomerative complete linkage algorithm and the diameter k -clustering problem. However, since their model differs from ours, their results have no bearing on our results.

1.2 Our Contribution

In this paper, we study the agglomerative complete linkage clustering algorithm and related algorithms for input sets $X \subset \mathbb{R}^d$. To measure the distance between data points, we use a metric that is based on a norm, e.g., the Euclidean metric. We prove

that the agglomerative solution to the diameter k -clustering problem is an $O(\log k)$ -approximation. Here, the O -notation hides a constant that is doubly exponential in d . This approximation guarantee holds for every level of the hierarchy computed by the algorithm. That is, we compare each computed k -clustering with an optimal solution for that particular value of k . These optimal k -clusterings do not necessarily form a hierarchy. In fact, there are simple examples where optimal solutions have no hierarchical structure.

Our analysis also yields that if we allow $2k$ instead of k clusters and compare the cost of the computed $2k$ -clustering to an optimal solution with k clusters, the approximation factor is independent of k and depends only on d . Moreover, the techniques of our analysis can be applied to prove stronger results for the k -center problem and the discrete k -center problem. For the k -center problem, we derive an approximation guarantee that is logarithmic in k and only singly exponential in d . For the discrete k -center problem, we derive an approximation guarantee that is logarithmic in k and the dependence on d is only linear and additive.

Furthermore, we give almost matching upper and lower bounds for the one-dimensional case. These bounds are independent of k . For $d \geq 2$ and the metric based on the ℓ_∞ -norm, we provide a lower bound that exceeds the upper bound for $d = 1$. For $d \geq 3$, we give a lower bound for the Euclidean case which is larger than the lower bound for $d = 1$. Finally, we construct instances providing lower bounds for any metric based on an ℓ_p -norm with $1 \leq p \leq \infty$. However, the construction of these instances needs the dimension to depend on k .

2 Preliminaries and Problem Definitions

Throughout this paper, we consider input sets that are finite subsets of \mathbb{R}^d . Our results hold for arbitrary metrics that are based on a norm, i.e., the distance $\|x - y\|$ between two points $x, y \in \mathbb{R}^d$ is measured using an arbitrary norm $\|\cdot\|$. Readers who are not familiar with arbitrary metrics or are only interested in the Euclidean case, may assume that $\|\cdot\|_2$ is used, i.e. $\|x - y\| = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$. For $r \in \mathbb{R}$ and $y \in \mathbb{R}^d$, we denote the closed d -dimensional ball of radius r centered at y by $B_r^d(y) := \{x \mid \|x - y\| \leq r\}$.

Given $k \in \mathbb{N}$ and a finite set $X \subset \mathbb{R}^d$ with $k \leq |X|$, we say that $\mathcal{C}_k = \{C_1, \dots, C_k\}$ is a k -clustering of X if the sets C_1, \dots, C_k (called clusters) form a partition of X into k non-empty subsets. We call a collection of k -clusterings of the same finite set X but for different values of k hierarchical, if it fulfills the following two properties. First, for any $1 \leq k \leq |X|$ the collection contains at most one k -clustering. Second, for any two of its clusterings $\mathcal{C}_i, \mathcal{C}_j$ with $|\mathcal{C}_i| = i < j = |\mathcal{C}_j|$, every cluster in \mathcal{C}_i is the union of one or more clusters from \mathcal{C}_j . A hierarchical collection of clusterings is called a hierarchical clustering.

We define the diameter of a finite and non-empty set $C \subset \mathbb{R}^d$ to be $\text{diam}(C) := \max_{x, y \in C} \|x - y\|$. Furthermore, we define the diameter cost of a k -clustering \mathcal{C}_k as its largest diameter, i.e. $\text{cost}_{\text{diam}}(\mathcal{C}_k) := \max_{C \in \mathcal{C}_k} \text{diam}(C)$. The radius of C is defined as $\text{rad}(C) := \min_{y \in \mathbb{R}^d} \max_{x \in C} \|x - y\|$ and the radius cost of a k -clustering \mathcal{C}_k is defined as its largest radius, i.e. $\text{cost}_{\text{rad}}(\mathcal{C}_k) := \max_{C \in \mathcal{C}_k} \text{rad}(C)$. Finally, we define the

discrete radius of C to be $\text{drad}(C) := \min_{y \in C} \max_{x \in C} \|x - y\|$ and the discrete radius cost of a k -clustering \mathcal{C}_k is defined as its largest discrete radius, i.e. $\text{cost}_{\text{drad}}(\mathcal{C}_k) := \max_{C \in \mathcal{C}_k} \text{drad}(C)$.

Problem 1 (Discrete k -Center) Given $k \in \mathbb{N}$ and a finite set $X \subset \mathbb{R}^d$ with $|X| \geq k$, find a k -clustering \mathcal{C}_k of X with minimal discrete radius cost.

Problem 2 (k -Center) Given $k \in \mathbb{N}$ and a finite set $X \subset \mathbb{R}^d$ with $|X| \geq k$, find a k -clustering \mathcal{C}_k of X with minimal radius cost.

Problem 3 (Diameter k -Clustering) Given $k \in \mathbb{N}$ and a finite set $X \subset \mathbb{R}^d$ with $|X| \geq k$, find a k -clustering \mathcal{C}_k of X with minimal diameter cost.

For our analysis of agglomerative clustering, we repeatedly use the volume argument stated in Lemma 5. This argument provides an upper bound on the minimum distance between two points from a finite set of points lying inside the union of finitely many balls. For the application of this argument, the following definition is crucial.

Definition 4 Let $k \in \mathbb{N}$ and $r \in \mathbb{R}$. A set $X \subset \mathbb{R}^d$ is called (k, r) -coverable if there exist $y_1, \dots, y_k \in \mathbb{R}^d$ with $X \subseteq \bigcup_{i=1}^k B_r^d(y_i)$.

Lemma 5 Let $k \in \mathbb{N}$, $r \in \mathbb{R}$ and $P \subset \mathbb{R}^d$ be finite and (k, r) -coverable with $|P| > k$. Then, there exist distinct $p, q \in P$ such that $\|p - q\| \leq 4r \sqrt{\frac{k}{|P|}}$.

Proof Let $Z \subset \mathbb{R}^d$ with $|Z| = k$ and $P \subset \bigcup_{z \in Z} B_r^d(z)$. We define δ to be the minimum distance between two points of P , i.e. $\delta := \min_{\substack{p, q \in P \\ p \neq q}} \|p - q\|$. We assume

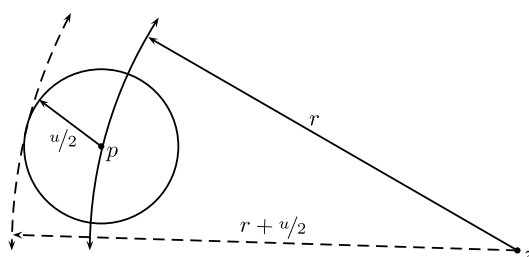
for contradiction that $u := 4r \sqrt{\frac{k}{|P|}} < \delta$. Since $|P| > k$ there exists $z \in Z$ with $|B_r^d(z) \cap P| \geq 2$. It follows that $\delta \leq 2r$ and hence, $\frac{u}{2} < r$. Note that for any $y \in \mathbb{R}^d$, $R \in \mathbb{R}$, and any norm $\|\cdot\|$, we have $\text{vol}(B_R^d(y)) = R^d \cdot V_d$, where V_d is the volume of the d -dimensional unit ball $B_1^d(0)$ (see [16], Corollary 6.2.15). Therefore, we deduce

$$\text{vol}\left(\bigcup_{z \in Z} B_{r+\frac{u}{2}}^d(z)\right) < \sum_{z \in Z} \text{vol}(B_{2r}^d(z)) \leq k \cdot (2r)^d \cdot V_d.$$

Furthermore, since any $p \in P$ is contained in a ball $B_r^d(z)$ for some $z \in Z$, we conclude that any ball $B_{\frac{u}{2}}^d(p)$ for $p \in P$ is contained in a ball $B_{r+\frac{u}{2}}^d(z)$ for some $z \in Z$ (see Fig. 1). Thus, we deduce

$$\text{vol}\left(\bigcup_{p \in P} B_{\frac{u}{2}}^d(p)\right) < k \cdot (2r)^d \cdot V_d. \quad (1)$$

Fig. 1 The volume argument



However, since $u < \delta$, for any distinct $p, q \in P$, we have $B_{\frac{u}{2}}^d(p) \cap B_{\frac{u}{2}}^d(q) = \emptyset$. Therefore, the total volume of the $|P|$ balls $B_{\frac{u}{2}}^d(p)$ is given by

$$\text{vol}\left(\bigcup_{p \in P} B_{\frac{u}{2}}^d(p)\right) = |P| \left(\frac{u}{2}\right)^d V_d = k \cdot (2r)^d \cdot V_d,$$

using the definition of u . This contradicts (1). We obtain $\delta \leq u$, which proves the lemma. \square

3 Analysis

In this section we analyze the agglomerative clustering algorithms for the (discrete) k -center problem and the diameter k -clustering problem. As mentioned in the introduction, an agglomerative algorithm takes a bottom-up approach. It starts with the $|X|$ -clustering that contains one cluster for each input point and then successively merges two of the remaining clusters such that the cost of the resulting clustering is minimized. That is, in each merge step the agglomerative algorithms for Problems 1, 2 and 3 minimize the discrete radius, the radius and the diameter of the resulting cluster, respectively.

Our main objective is the agglomerative complete linkage clustering algorithm, which minimizes the diameter in every step. Nevertheless, we start with the analysis of the agglomerative algorithm for the discrete k -center problem since it is the simplest one of the three. Then we adapt our analysis to the k -center problem and finally to the diameter k -clustering problem. In each case we need to introduce further techniques to deal with the increased complexity of the given problem.

We show that all three algorithms compute an $O(\log k)$ -approximation for the particular corresponding clustering problem. However, the dependency on the dimension which is hidden in the O -notation ranges from only linear and additive in case of the discrete k -center problem to a factor that is doubly exponential in case of the diameter k -clustering problem.

As mentioned in the introduction, the cost of optimal solutions to the three problems are within a factor of 2 from each other. That is, each algorithm computes an $O(\log k)$ -approximation for all three problems. However, we will analyze the proper agglomerative algorithm for each problem.

AGGLOMERATIVEDISCRETERADIUS(X):
 X finite set of input points from \mathbb{R}^d

```

1:  $\mathcal{C}_{|X|} := \{\{x\} \mid x \in X\}$ 
2: for  $i = |X| - 1, \dots, 1$  do
3:   find distinct clusters  $A, B \in \mathcal{C}_{i+1}$  minimizing  $\text{drad}(A \cup B)$ 
4:    $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$ 
5: end for
6: return  $\mathcal{C}_1, \dots, \mathcal{C}_{|X|}$ 

```

Algorithm 1: The agglomerative algorithm for the discrete k -center problem

3.1 Discrete k -Center Clustering

The agglomerative algorithm for the discrete k -center problem is stated as Algorithm 1. Given a finite set $X \subset \mathbb{R}^d$ of input points, the algorithm computes hierarchical k -clusterings for all values of k between 1 and $|X|$. We denote them by $\mathcal{C}_1, \dots, \mathcal{C}_{|X|}$. Throughout this section, *cost* always means discrete radius cost. opt_k refers to the cost of an optimal discrete k -center clustering of $X \subset \mathbb{R}^d$ where $k \in \mathbb{N}$ with $k \leq |X|$, i.e. the cost of an optimal solution to Problem 1.

The following theorem states our result for the discrete k -center problem.

Theorem 6 *Let $X \subset \mathbb{R}^d$ be a finite set of points. Then, for all $k \in \mathbb{N}$ with $k \leq |X|$, the partition \mathcal{C}_k of X into k clusters as computed by Algorithm 1 satisfies*

$$\text{cost}_{\text{drad}}(\mathcal{C}_k) < (20d + 2 \log_2(k) + 2) \cdot \text{opt}_k,$$

where opt_k denotes the cost of an optimal solution to Problem 1.

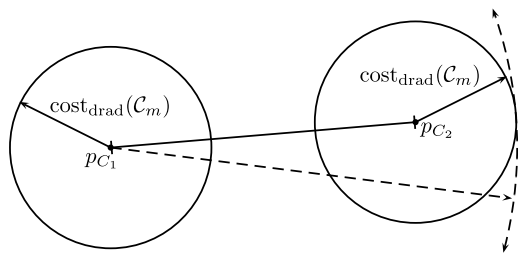
Since any cluster C is contained in a ball of radius $\text{drad}(C)$, we have that X is $(k, \text{cost}_{\text{drad}}(\mathcal{C}_k))$ -coverable for any k -clustering \mathcal{C}_k of X . It follows, that X is (k, opt_k) -coverable. This fact, as well as the following observation about the greedy strategy of Algorithm 1, will be used frequently in our analysis..

Observation 7 *The cost of all computed clusterings is equal to the discrete radius of the cluster created last. Furthermore, the discrete radius of the union of any two clusters is always an upper bound for the cost of the clustering to be computed next.*

We prove Theorem 6 in two steps. First, Proposition 8 in Sect. 3.1.1 provides an upper bound to the cost of the intermediate $2k$ -clustering. This upper bound is independent of k and $|X|$, only linear in d and may be of independent interest. In its proof, we use Lemma 5 to bound the distance between the centers of pairs of remaining clusters. The cost of merging such a pair gives an upper bound to the cost of the next merge step. Therefore, we can bound the discrete radius of the created cluster by the sum of the larger of the two clusters' discrete radii and the distance between their centers.

Second, in Sect. 3.1.2, we analyze the remaining k merge steps of Algorithm 1 down to the computation of the k -clustering. There, we no longer need to apply the

Fig. 2 $\text{drad}(C_1 \cup C_2) < \text{cost}_{\text{drad}}(C_m) + \|p_{C_1} - p_{C_2}\|$



volume argument from Lemma 5 to bound the distance between two cluster centers. It will be replaced by a very simple bound that is already sufficient. Analogously to the first step, this leads to a bound for the cost of merging a pair of clusters.

3.1.1 Analysis of the $2k$ -Clustering

Proposition 8 *Let $X \subset \mathbb{R}^d$ be finite. Then, for all $k \in \mathbb{N}$ with $2k \leq |X|$, the partition C_{2k} of X into $2k$ clusters as computed by Algorithm 1 satisfies*

$$\text{cost}_{\text{drad}}(C_{2k}) < 20d \cdot \text{opt}_k,$$

where opt_k denotes the cost of an optimal solution to Problem 1.

To prove Proposition 8, we divide the merge steps of Algorithm 1 into phases, each reducing the number of remaining clusters by one fourth. The following lemma bounds the increase of the cost during a single phase by an additive term.

Lemma 9 *Let $m \in \mathbb{N}$ with $2k < m \leq |X|$. Then,*

$$\text{cost}_{\text{drad}}(C_{\lfloor \frac{3m}{4} \rfloor}) < \text{cost}_{\text{drad}}(C_m) + 4\sqrt{\frac{2k}{m}} \cdot \text{opt}_k.$$

Proof Let $R := \text{cost}_{\text{drad}}(C_m)$. From every cluster $C \in C_m$, we fix a center $p_C \in C$ with $C \subset B_R^d(p_C)$.

Let $t := \lfloor \frac{3m}{4} \rfloor$. Then, $C_m \cap C_{t+1}$ is the set of clusters from C_m that still exist $\lceil \frac{m}{4} \rceil - 1$ merge steps after the computation of C_m . In each iteration of its loop, the algorithm can merge at most two clusters from C_m . Thus, $|C_m \cap C_{t+1}| > \frac{m}{2}$.

Let $P := \{p_C \mid C \in C_m \cap C_{t+1}\}$. Then, $|P| = |C_m \cap C_{t+1}| > \frac{m}{2} > k$. Since X is (k, opt_k) -coverable, so is $P \subset X$. Therefore, by Lemma 5, there exist distinct $C_1, C_2 \in C_m \cap C_{t+1}$ such that $\|p_{C_1} - p_{C_2}\| \leq 4\sqrt{\frac{2k}{m}} \cdot \text{opt}_k$. Then, the distance from p_{C_1} to any $q \in C_2$ is at most $4\sqrt{\frac{2k}{m}} \cdot \text{opt}_k + R$. We conclude that merging C_1 and C_2 would result in a cluster whose discrete radius can be upper bounded by $\text{drad}(C_1 \cup C_2) < \text{cost}_{\text{drad}}(C_m) + 4\sqrt{\frac{2k}{m}} \cdot \text{opt}_k$ (see Fig. 2). The result follows using $C_1, C_2 \in C_{t+1}$ and Observation 7. \square

To prove Proposition 8, we apply Lemma 9 for $\lceil \log_{\frac{4}{3}} \frac{|X|}{2k} \rceil$ consecutive phases.

Proof of Proposition 8 Let $u := \lceil \log_{\frac{4}{3}} \frac{|X|}{2k} \rceil$ and define $m_i := \lceil (\frac{3}{4})^i |X| \rceil$ for all $i = 0, \dots, u$. Then, $m_u \leq 2k$ and $m_i > 2k$ for all $i = 0, \dots, u - 1$. Since $\lfloor \frac{3m_i}{4} \rfloor = \lfloor \frac{3}{4} \lceil (\frac{3}{4})^i |X| \rceil \rfloor \leq \lfloor (\frac{3}{4})^{i+1} |X| + \frac{3}{4} \rfloor \leq \lceil (\frac{3}{4})^{i+1} |X| \rceil = m_{i+1}$ and Algorithm 1 uses a greedy strategy, we get $\text{cost}_{\text{drad}}(C_{m_{i+1}}) \leq \text{cost}_{\text{drad}}(C_{\lfloor \frac{3m_i}{4} \rfloor})$ for all $i = 0, \dots, u - 1$. Combining this with Lemma 9 (applied to $m = m_i$), we obtain

$$\text{cost}_{\text{drad}}(C_{m_{i+1}}) < \text{cost}_{\text{drad}}(C_{m_i}) + 4 \sqrt{\frac{2k}{m_i}} \cdot \text{opt}_k.$$

By repeatedly applying this inequality for $i = 0, \dots, u - 1$ and using $\text{cost}_{\text{drad}}(C_{2k}) \leq \text{cost}_{\text{drad}}(C_{m_u})$ and $\text{cost}_{\text{drad}}(C_{m_0}) = 0$, we deduce

$$\text{cost}_{\text{drad}}(C_{2k}) < \sum_{i=0}^{u-1} \left(4 \sqrt{\frac{2k}{m_i}} \cdot \text{opt}_k \right) < 4 \sqrt{\frac{2k}{|X|}} \cdot \sum_{i=0}^{u-1} \sqrt{\left(\frac{4}{3}\right)^i} \cdot \text{opt}_k.$$

Solving the geometric series and using $u - 1 < \log_{\frac{4}{3}} \frac{|X|}{2k}$ leads to

$$\text{cost}_{\text{drad}}(C_{2k}) < 4 \sqrt{\frac{2k}{|X|}} \cdot \frac{\sqrt{\left(\frac{4}{3}\right)^u} - 1}{\sqrt{\frac{4}{3}} - 1} \cdot \text{opt}_k < \frac{4 \sqrt{\frac{4}{3}}}{\sqrt{\frac{4}{3}} - 1} \cdot \text{opt}_k. \quad (2)$$

By taking only the first two terms of the series expansion of the exponential function, we get $\sqrt{\frac{4}{3}} = e^{\frac{\ln \frac{4}{3}}{2}} > 1 + \frac{\ln \frac{4}{3}}{2}$. Substituting this bound into (2) gives

$$\text{cost}_{\text{drad}}(C_{2k}) < \frac{4 \sqrt{\frac{4}{3}}}{\ln \frac{4}{3}} d \cdot \text{opt}_k < 20d \cdot \text{opt}_k. \quad \square$$

3.1.2 Analysis of the Remaining Merge Steps

The analysis of the remaining merge steps introduces the $O(\log k)$ term to the approximation factor of our result. It is similar to the analysis used in the proof of Proposition 8. Again, we divide the merge steps into phases. However, this time one phase consists of one half of the remaining merge steps. Furthermore, we are able to replace the volume argument from Lemma 5 by a simpler bound. More precisely, as long as there are more than k clusters left, we are able to find a pair of clusters whose centers lie in the same cluster of an optimal k -clustering. That is, the distance between the centers is at most two times the discrete radius of the common cluster in the optimal clustering. The following lemma bounds the increase of the cost during a single phase.

Lemma 10 *Let $m \in \mathbb{N}$ with $k < m \leq |X|$. Then,*

$$\text{cost}_{\text{drad}}(C_{k + \lfloor \frac{m-k}{2} \rfloor}) < \text{cost}_{\text{drad}}(C_m) + 2 \text{opt}_k.$$

Proof Let $R := \text{cost}_{\text{drad}}(\mathcal{C}_m)$. From every cluster $C \in \mathcal{C}_m$, we fix a center $p_C \in C$ with $C \subset B_R^d(p_C)$.

Let $t := k + \lfloor \frac{m-k}{2} \rfloor$. Then, $\mathcal{C}_m \cap \mathcal{C}_{t+1}$ is the set of clusters from \mathcal{C}_m that still exist $\lceil \frac{m-k}{2} \rceil - 1$ merge steps after the computation of \mathcal{C}_m . In each iteration of its loop, the algorithm can merge at most two clusters from \mathcal{C}_m . Thus, $|\mathcal{C}_m \cap \mathcal{C}_{t+1}| > k$.

Let $P := \{p_C \mid C \in \mathcal{C}_m \cap \mathcal{C}_{t+1}\}$. Since X is (k, opt_k) -coverable, so is $P \subset X$. Therefore, using $|P| > k$ it follows that there exist distinct $C_1, C_2 \in \mathcal{C}_m \cap \mathcal{C}_{t+1}$ such that p_{C_1} and p_{C_2} are contained in the same ball of radius opt_k , i.e. $\|p_{C_1} - p_{C_2}\| \leq 2 \text{opt}_k$. Then, the distance from p_{C_1} to any $q \in C_2$ is at most $2 \text{opt}_k + R$. We conclude that merging C_1 and C_2 would result in a cluster whose discrete radius can be upper bounded by $\text{drad}(C_1 \cup C_2) < \text{cost}_{\text{drad}}(\mathcal{C}_m) + 2 \text{opt}_k$ (see Fig. 2). The result follows using $C_1, C_2 \in \mathcal{C}_{t+1}$ and Observation 7. \square

To prove Theorem 6, we apply Lemma 10 for about $\log k$ consecutive phases.

Proof of Theorem 6 Let $\varepsilon > 0$ and $u := \lceil \log_2(k) + \varepsilon \rceil$ such that $\log_2 k < u \leq \log_2(k) + 1$. Furthermore, define $m_i := k + \lfloor (\frac{1}{2})^i k \rfloor$ for all $i = 0, \dots, u$. Then, $m_u = k$ and $m_i > k$ for all $i = 0, \dots, u - 1$. Since $k + \lfloor \frac{m_i - k}{2} \rfloor = k + \lfloor \frac{1}{2} \lfloor (\frac{1}{2})^i k \rfloor \rfloor \leq k + \lfloor (\frac{1}{2})^{i+1} k \rfloor = m_{i+1}$ and Algorithm 1 uses a greedy strategy, we deduce $\text{cost}_{\text{drad}}(\mathcal{C}_{m_{i+1}}) \leq \text{cost}_{\text{drad}}(\mathcal{C}_{k + \lfloor \frac{m_i - k}{2} \rfloor})$ for all $i = 0, \dots, u - 1$. Combining this with Lemma 10 (applied to $m = m_i$), we obtain

$$\text{cost}_{\text{diam}}(\mathcal{C}_{m_{i+1}}) < \text{cost}_{\text{drad}}(\mathcal{C}_{m_i}) + 2 \text{opt}_k.$$

By repeatedly applying this inequality for $i = 0, \dots, u - 1$ and using $m_0 = 2k$, we get

$$\text{cost}_{\text{drad}}(\mathcal{C}_k) < \text{cost}_{\text{drad}}(\mathcal{C}_{2k}) + \sum_{i=0}^{u-1} 2 \text{opt}_k \leq \text{cost}_{\text{drad}}(\mathcal{C}_{2k}) + (2 \log_2(k) + 2) \cdot \text{opt}_k.$$

Hence, the result follows using Proposition 8. \square

3.2 k -Center Clustering

The agglomerative algorithm for the k -center problem is stated as Algorithm 2. The only difference to Algorithm 1 is the minimization of the radius instead of the discrete radius in Step 3.

In the following, *cost* always means radius cost and opt_k refers to the cost of an optimal k -center clustering of $X \subset \mathbb{R}^d$ where $k \in \mathbb{N}$ with $k \leq |X|$.

Observation 11 (Analogous to Observation 7) *The cost of all computed clusterings is equal to the radius of the cluster created last. Furthermore, the radius of the union of any two clusters is always an upper bound for the cost of the clustering to be computed next.*

The following theorem states our result for the k -center problem.

AGGLOMERATERADIUS(X):

X finite set of input points from \mathbb{R}^d

```

1:  $\mathcal{C}_{|X|} := \{\{x\} \mid x \in X\}$ 
2: for  $i = |X| - 1, \dots, 1$  do
3:   find distinct clusters  $A, B \in \mathcal{C}_{i+1}$  minimizing  $\text{rad}(A \cup B)$ 
4:    $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$ 
5: end for
6: return  $\mathcal{C}_1, \dots, \mathcal{C}_{|X|}$ 

```

Algorithm 2: The agglomerative algorithm for the k -center problem.

Theorem 12 *Let $X \subset \mathbb{R}^d$ be a finite set of points. Then, for all $k \in \mathbb{N}$ with $k \leq |X|$, the partition \mathcal{C}_k of X into k clusters as computed by Algorithm 2 satisfies*

$$\text{cost}_{\text{rad}}(\mathcal{C}_k) = O(\log k) \cdot \text{opt}_k,$$

where opt_k denotes the cost of an optimal solution to Problem 2, and the constant hidden in the O -notation is singly exponential in the dimension d .

Theorem 12 holds for any particular tie-breaking strategy. However, to keep the analysis simple, we assume that there are no ties. That is, we assume that for any input set X the clusterings computed by Algorithm 2 are uniquely determined. As in the proof of Theorem 6, we first show a bound for the cost of the intermediate $2k$ -clustering. However, we have to apply a different analysis. As a consequence, the dependency on the dimension increases from linear and additive to a singly exponential factor.

3.2.1 Analysis of the $2k$ -Clustering

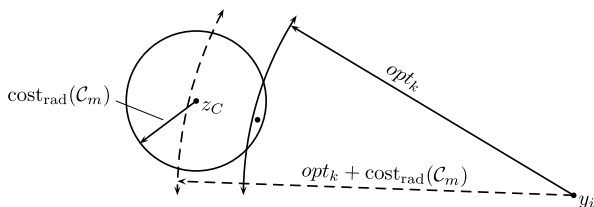
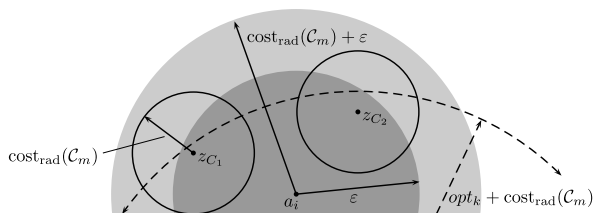
Proposition 13 *Let $X \subset \mathbb{R}^d$ be finite. Then, for all $k \in \mathbb{N}$ with $2k \leq |X|$, the partition \mathcal{C}_{2k} of X into $2k$ clusters as computed by Algorithm 2 satisfies*

$$\text{cost}_{\text{rad}}(\mathcal{C}_{2k}) < 24d \cdot e^{24d} \cdot \text{opt}_k,$$

where opt_k denotes the cost of an optimal solution to Problem 2.

Just as in the analysis of Algorithm 1, we divide the merge steps of Algorithm 2 into phases, such that in each phase the number of remaining clusters is reduced by one fourth. Like in the discrete case, the input points are (k, opt_k) -coverable. However, centers corresponding to an intermediate solution computed by Algorithm 2 need not be covered by the k balls induced by an optimal solution. As a consequence, we are no longer able to apply Lemma 5 on the centers as in the discrete case.

To bound the increase of the cost during a single phase, we cover the remaining clusters at the beginning of a phase by a set of overlapping balls. Each of the clusters is completely contained in one of these balls that all have the same radius. Furthermore, the number of remaining clusters will be at least twice the number of these balls. It follows that there are many pairs of clusters that are contained in the

Fig. 3 Intermediate centers

Fig. 4 Covering centers and clusters


same ball. Then, as long as the existence of at least one such pair can be guaranteed, the radius of the cluster created next can be bounded by the radius of the balls. The following lemma will be used to bound the increase of the cost during a single phase.

Lemma 14 *Let $m \in \mathbb{N}$ with $2k < m \leq |X|$. Then,*

$$\text{cost}_{\text{rad}}(C_{\lfloor \frac{3m}{4} \rfloor}) < \left(1 + 6\sqrt{\frac{2k}{m}}\right) \cdot \text{cost}_{\text{rad}}(C_m) + 6\sqrt{\frac{2k}{m}} \cdot \text{opt}_k.$$

Proof Let $\mathcal{P} = \{P_1, \dots, P_k\}$ be an optimal k -clustering of X . We fix $y_1, \dots, y_k \in \mathbb{R}^d$ such that $P_i \subset B_{\text{opt}_k}(y_i)$ for $i = 1, \dots, k$. For any $C \in C_m$ let $z_C \in \mathbb{R}^d$ such that $C \subset B_{\text{cost}_{\text{rad}}(C_m)}(z_C)$. It follows that each z_C is contained in at least one of the balls $B_{\text{opt}_k + \text{cost}_{\text{rad}}(C_m)}(y_i)$ for $i = 1, \dots, k$ (see Fig. 3).

For $\lambda \in \mathbb{R}$ with $\lambda > 0$ a ball of radius $\text{opt}_k + \text{cost}_{\text{rad}}(C_m)$ can be covered by $\lceil (\frac{3}{\lambda})^d \rceil$ balls of radius $\lambda(\text{opt}_k + \text{cost}_{\text{rad}}(C_m))$ (see [13]). Choosing $\lambda = \frac{3}{\sqrt{\lfloor \frac{m}{2k} \rfloor}}$, we get that each of the balls $B_{\text{opt}_k + \text{cost}_{\text{rad}}(C_m)}(y_i)$ for $i = 1, \dots, k$ can be covered by $\ell := \lfloor \frac{m}{2k} \rfloor \leq \frac{m}{2k}$ balls of radius $\varepsilon = \frac{3}{\sqrt{\lfloor \frac{m}{2k} \rfloor}}(\text{opt}_k + \text{cost}_{\text{rad}}(C_m))$. Therefore, there exist $k \cdot \ell \leq \frac{m}{2}$ balls $B_1, \dots, B_{k\ell}$ of radius ε such that each z_C for $C \in C_m$ is contained in at least one of these balls. For $i = 1, \dots, k\ell$ let $a_i \in \mathbb{R}^d$ such that $B_i = B_\varepsilon(a_i)$. Then, any cluster $C \in C_m$ is contained in at least one of the balls $A_1, \dots, A_{k\ell}$ with $A_i = B_{\text{cost}_{\text{rad}}(C_m) + \varepsilon}(a_i)$ for $i = 1, \dots, k\ell$ (see Fig. 4).

Let $t := \lfloor \frac{3m}{4} \rfloor$ and $C_m \cap C_{t+1}$ be the set of clusters from C_m that still exist $\lceil \frac{m}{4} \rceil - 1$ merge steps after the computation of C_m . In each iteration of its loop, the algorithm can merge at most two clusters from C_m . Thus, $|C_m \cap C_{t+1}| > \frac{m}{2}$.

Since $k\ell \leq \frac{m}{2}$, there exist two clusters $C_1, C_2 \in C_m \cap C_{t+1}$ that are contained in the same ball A_i with $i \in \{1, \dots, k\ell\}$. Therefore, merging clusters C_1 and C_2 would result in a cluster whose radius can be upper bounded by $\text{rad}(C_1 \cup C_2) \leq \text{cost}_{\text{rad}}(C_m) + \varepsilon$. Using Observation 11 and the fact that C_1 and C_2 are part of the clustering C_{t+1} , we

can upper bound the cost of \mathcal{C}_t by

$$\text{cost}_{\text{rad}}(\mathcal{C}_t) \leq \text{cost}_{\text{rad}}(\mathcal{C}_m) + \varepsilon.$$

It remains to show $\varepsilon < 6\sqrt[d]{\frac{2k}{m}} \cdot (\text{opt}_k + \text{cost}_{\text{rad}}(\mathcal{C}_m))$. Since $\frac{m}{2k} > 1$, we have $\frac{m}{2k} < 2\lfloor \frac{m}{2k} \rfloor$. Thus, $\sqrt[d]{\frac{m}{2k}} < \sqrt[d]{2\lfloor \frac{m}{2k} \rfloor} \leq 2\sqrt[d]{\lfloor \frac{m}{2k} \rfloor}$ and $\frac{3}{\sqrt[d]{\lfloor \frac{m}{2k} \rfloor}} < 6\sqrt[d]{\frac{2k}{m}}$. \square

To prove Proposition 13, we apply Lemma 14 for $\lceil \log_{\frac{4}{3}} \frac{|X|}{2k} \rceil$ consecutive phases.

Proof of Proposition 13 Let $u := \lceil \log_{\frac{4}{3}} \frac{|X|}{2k} \rceil$ and define $m_i := \lceil (\frac{3}{4})^i |X| \rceil$ for all $i = 0, \dots, u$. Then, $m_u \leq 2k$ and $m_i > 2k$ for all $i = 0, \dots, u-1$. Analogously to the proof of Proposition 8, we get $\lfloor \frac{3m_i}{4} \rfloor \leq m_{i+1}$ and using Lemma 14, we deduce $\text{cost}_{\text{rad}}(\mathcal{C}_{m_{i+1}}) < (1 + 6\sqrt[d]{\frac{2k}{m_i}}) \cdot \text{cost}_{\text{rad}}(\mathcal{C}_{m_i}) + 6\sqrt[d]{\frac{2k}{m_i}} \cdot \text{opt}_k$ for all $i = 0, \dots, u-1$. By repeatedly applying this inequality and using $\text{cost}_{\text{rad}}(\mathcal{C}_{2k}) \leq \text{cost}_{\text{rad}}(\mathcal{C}_{m_u})$ and $\text{cost}_{\text{rad}}(\mathcal{C}_{m_0}) = 0$, we get

$$\begin{aligned} & \text{cost}_{\text{rad}}(\mathcal{C}_{2k}) \\ & < \sum_{i=0}^{u-1} \left(6\sqrt[d]{\frac{2k}{m_i}} \cdot \prod_{j=i+1}^{u-1} \left(1 + 6\sqrt[d]{\frac{2k}{m_j}} \right) \right) \cdot \text{opt}_k \\ & \leq 6\sqrt[d]{\frac{2k}{|X|}} \cdot \sum_{i=0}^{u-1} \left(\left(\frac{4}{3} \right)^{\frac{i}{d}} \cdot \prod_{j=i+1}^{u-1} \left(1 + 6\sqrt[d]{\frac{2k}{|X|}} \cdot \left(\frac{4}{3} \right)^{\frac{j}{d}} \right) \right) \cdot \text{opt}_k \quad (3) \\ & = 6\sqrt[d]{\frac{2k}{|X|}} \left(\frac{4}{3} \right)^{\frac{u-1}{d}} \\ & \quad \cdot \sum_{i=0}^{u-1} \left(\left(\frac{3}{4} \right)^{\frac{i}{d}} \cdot \prod_{j=u-i}^{u-1} \left(1 + 6\sqrt[d]{\frac{2k}{|X|}} \cdot \left(\frac{4}{3} \right)^{\frac{u-1}{d}} \left(\frac{3}{4} \right)^{\frac{u-1-j}{d}} \right) \right) \cdot \text{opt}_k. \quad (4) \end{aligned}$$

Here, we obtain (3) using $m_i \geq (\frac{3}{4})^i |X|$ and we obtain (4) by substituting $u-1-i$ for i . Using $u-1 < \log_{\frac{4}{3}} \frac{|X|}{2k}$, we deduce

$$\text{cost}_{\text{rad}}(\mathcal{C}_{2k}) < 6 \sum_{i=0}^{u-1} \left(\left(\frac{3}{4} \right)^{\frac{i}{d}} \cdot \prod_{j=0}^{i-1} \left(1 + 6 \left(\frac{3}{4} \right)^{\frac{j}{d}} \right) \right) \cdot \text{opt}_k. \quad (5)$$

By taking only the first two terms of the series expansion of the exponential function, we get $1 + 6 \left(\frac{3}{4} \right)^{\frac{j}{d}} < e^{6 \left(\frac{3}{4} \right)^{\frac{j}{d}}}$ and therefore

$$\prod_{j=0}^{i-1} \left(1 + 6 \left(\frac{3}{4} \right)^{\frac{j}{d}} \right) < \prod_{j=0}^{i-1} \left(e^{6 \left(\frac{3}{4} \right)^{\frac{j}{d}}} \right) = e^{6 \sum_{j=0}^{i-1} \left(\frac{3}{4} \right)^{\frac{j}{d}}}. \quad (6)$$

The sum in the exponent can be bounded by the infinite geometric series

$$\sum_{j=0}^{\infty} \left(\frac{3}{4}\right)^{\frac{j}{d}} < \frac{1}{1 - (\frac{3}{4})^{\frac{1}{d}}} \leq 4d, \quad (7)$$

where the last inequality follows by upper bounding the convex function $f(x) = (\frac{3}{4})^x$ in the interval $[0, 1]$ by the line through $f(0)$ and $f(1)$. Putting Inequalities (5), (6) and (7) together then gives

$$\text{cost}_{\text{rad}}(\mathcal{C}_{2k}) < 6 \sum_{i=0}^{u-1} \left(\left(\frac{3}{4}\right)^{\frac{i}{d}} \cdot e^{24d} \right) \cdot \text{opt}_k < 24d \cdot e^{24d} \cdot \text{opt}_k,$$

where the last inequality follows by using Inequality (7) again. \square

3.2.2 Connected Instances

The analysis of the remaining merge steps from the discrete k -center case (cf. Sect. 3.1.2) is not transferable to the k -center case. Again, as in the proof of Proposition 13, we are no longer able to derive a simple additive bound on the increase of the cost when merging two clusters. In order to preserve the logarithmic dependency of the approximation factor on k , we show that it is sufficient to analyze Algorithm 2 on a subset $Y \subseteq X$ satisfying a certain connectivity property. Using this property, we are able to apply a combinatorial approach that relies on the number of merge steps left.

We start by defining the connectivity property that will be used to relate clusters to an optimal k -clustering.

Definition 15 Let $Z \subseteq \mathbb{R}^d$ and $r \in \mathbb{R}$. Two sets $A, B \subseteq \mathbb{R}^d$ are called (Z, r) -connected if there exists a $z \in Z$ with $B_r^d(z) \cap A \neq \emptyset$ and $B_r^d(z) \cap B \neq \emptyset$.

Note that for any two (Z, r) -connected clusters A, B , we have

$$\text{rad}(A \cup B) \leq \text{rad}(A) + 2 \cdot \text{rad}(B) + 2r. \quad (8)$$

Next, we show that for any input set X we can bound the cost of the k -clustering computed by Algorithm 2 by the cost of the ℓ -clustering computed by the algorithm on a connected subset $Y \subseteq X$ for a proper $\ell \leq k$. Recall that by our convention from the beginning of Sect. 3.2, the clusterings computed by Algorithm 2 on a particular input set are uniquely determined.

Lemma 16 Let $X \subset \mathbb{R}^d$ be finite and $k \in \mathbb{N}$ with $k \leq |X|$. Then, there exists a subset $Y \subseteq X$, a number $\ell \in \mathbb{N}$ with $\ell \leq k$ and $\ell \leq |Y|$, and a set $Z \subset \mathbb{R}^d$ with $|Z| = \ell$ such that:

1. Y is (ℓ, opt_k) -coverable;
2. $\text{cost}_{\text{rad}}(\mathcal{C}_k) \leq \text{cost}_{\text{rad}}(\mathcal{P}_\ell)$;

3. For all $n \in \mathbb{N}$ with $\ell + 1 \leq n \leq |Y|$, every cluster in \mathcal{P}_n is (Z, opt_k) -connected to another cluster in \mathcal{P}_n .

Here, the collection $\mathcal{P}_1, \dots, \mathcal{P}_{|Y|}$ denotes the hierarchical clustering computed by Algorithm 2 on input Y .

Proof To define Y , Z , and ℓ we consider the $(k + 1)$ -clustering computed by Algorithm 2 on input X . We know that $X = \bigcup_{A \in \mathcal{C}_{k+1}} A$ is (k, opt_k) -coverable. Let $E \subseteq \mathcal{C}_{k+1}$ be a minimal subset such that $\bigcup_{A \in E} A$ is $(|E| - 1, \text{opt}_k)$ -coverable, i.e., for all sets $F \subseteq \mathcal{C}_{k+1}$ with $|F| < |E|$ the union $\bigcup_{A \in F} A$ is not $(|F| - 1, \text{opt}_k)$ -coverable. Since a set F of size 1 cannot be $(|F| - 1, \text{opt}_k)$ -coverable, we get $|E| \geq 2$.

Let $Y := \bigcup_{A \in E} A$ and $\ell := |E| - 1$. Then, $\ell \leq k$ and Y is (ℓ, opt_k) -coverable. This establishes property 1.

It follows that there exists a set $Z \subset \mathbb{R}^d$ with $|Z| = \ell$ and $Y \subset \bigcup_{z \in Z} B_{\text{opt}_k}^d(z)$. Furthermore, we let $\mathcal{P}_1, \dots, \mathcal{P}_{|Y|}$ be the hierarchical clustering computed by Algorithm 2 on input Y .

Since Y is the union of the clusters from $E \subseteq \mathcal{C}_{k+1}$, each merge step between the computation of $\mathcal{C}_{|X|}$ and \mathcal{C}_{k+1} merges either two clusters $A, B \subset Y$ or two clusters $A, B \subset X \setminus Y$. The merge steps inside $X \setminus Y$ have no influence on the clusters inside Y . Furthermore, the merge steps inside Y would be the same in the absence of the clusters inside $X \setminus Y$. Therefore, on input Y , Algorithm 2 computes the $(\ell + 1)$ -clustering $\mathcal{P}_{\ell+1} = E = \mathcal{C}_{k+1} \cap 2^Y$. Thus, $\mathcal{P}_{\ell+1} \subseteq \mathcal{C}_{k+1}$.

To compute \mathcal{P}_ℓ , on input Y , Algorithm 2 merges two clusters from $\mathcal{P}_{\ell+1}$ that minimize the radius of the resulting cluster. Analogously, on input X , Algorithm 2 merges two clusters from \mathcal{C}_{k+1} to compute \mathcal{C}_k . Since $\mathcal{P}_{\ell+1} \subseteq \mathcal{C}_{k+1}$, Observation 11 implies $\text{cost}_{\text{rad}}(\mathcal{C}_k) \leq \text{cost}_{\text{rad}}(\mathcal{P}_\ell)$, thus proving property 2.

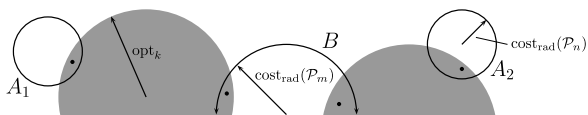
It remains to show that for all $n \in \mathbb{N}$ with $\ell + 1 \leq n \leq |Y|$ it holds that every cluster in \mathcal{P}_n is (Z, opt_k) -connected to another cluster in \mathcal{P}_n (property 3). By the definition of Z , every cluster in \mathcal{P}_n intersects at least one ball $B_{\text{opt}_k}^d(z)$ for $z \in Z$. Therefore, it is enough to show that each ball $B_{\text{opt}_k}^d(z)$ intersects at least two clusters from \mathcal{P}_n . We first show this property for $n = \ell + 1$. For $\ell = 1$ this follows from the fact that $B_{\text{opt}_k}^d(z)$ with $Z = \{z\}$ has to contain both clusters from \mathcal{P}_2 . For $\ell > 1$, we are otherwise able to remove one cluster from $\mathcal{P}_{\ell+1}$ and get ℓ clusters whose union is $(\ell - 1, \text{opt}_k)$ -coverable. This contradicts the definition of $E = \mathcal{P}_{\ell+1}$ as a minimal subset with this property.

To show property 3 for general n , let $C_1 \in \mathcal{P}_n$ and $z \in Z$ with $B_{\text{opt}_k}^d(z) \cap C_1 \neq \emptyset$. There exists a unique cluster $\tilde{C}_1 \in \mathcal{P}_{\ell+1}$ with $C_1 \subseteq \tilde{C}_1$. Then, we have $B_{\text{opt}_k}^d(z) \cap \tilde{C}_1 \neq \emptyset$. However, we have just shown that $B_{\text{opt}_k}^d(z)$ has to intersect at least two clusters from $\mathcal{P}_{\ell+1}$. Thus, there exists another cluster $\tilde{C}_2 \in \mathcal{P}_{\ell+1}$ with $B_{\text{opt}_k}^d(z) \cap \tilde{C}_2 \neq \emptyset$. Since every cluster from $\mathcal{P}_{\ell+1}$ is a union of clusters from \mathcal{P}_n , there exists at least one cluster $C_2 \in \mathcal{P}_n$ with $C_2 \subseteq \tilde{C}_2$ and $B_{\text{opt}_k}^d(z) \cap C_2 \neq \emptyset$. \square

3.2.3 Analysis of the Remaining Merge Steps

Let Y , Z , ℓ , and $\mathcal{P}_1, \dots, \mathcal{P}_{|Y|}$ be as given by Lemma 16. Then, Proposition 13 can be used to obtain an upper bound for the cost of $\mathcal{P}_{2\ell}$. In the following, we analyze the

Fig. 5 Merging
(Z, opt_k)-connected clusters



merge steps leading from $\mathcal{P}_{2\ell}$ to $\mathcal{P}_{\ell+1}$ and show how to obtain an upper bound for the cost of $\mathcal{P}_{\ell+1}$. As in Sect. 3.2.1, we analyze the merge steps in phases. The following lemma is used to bound the increase of the cost during a single phase. Note that opt_k still refers to the cost of an optimal solution on input X , not Y .

Lemma 17 *Let $m, n \in \mathbb{N}$ with $n \leq 2\ell$ and $\ell < m \leq n \leq |Y|$. If there are no two (Z, opt_k)-connected clusters in $\mathcal{P}_m \cap \mathcal{P}_n$, it holds*

$$\text{cost}_{\text{rad}}(\mathcal{P}_{\lfloor \frac{m+\ell}{2} \rfloor}) \leq \text{cost}_{\text{rad}}(\mathcal{P}_m) + 2 \cdot \text{cost}_{\text{rad}}(\mathcal{P}_n) + 2 \text{opt}_k.$$

Proof We show that there exist at least $m - \ell$ disjoint pairs of clusters from \mathcal{P}_m such that the radius of their union can be upper bounded by $\text{cost}_{\text{rad}}(\mathcal{P}_m) + 2 \cdot \text{cost}_{\text{rad}}(\mathcal{P}_n) + 2 \text{opt}_k$. By Observation 11, this upper bounds the cost of the computed clusterings as long as such a pair of clusters remains. Then, the lemma follows from the fact that in each iteration of its loop the algorithm can destroy at most two of these pairs. To bound the number of these pairs of clusters, we start with a structural observation. $\mathcal{P}_m \cap \mathcal{P}_n$ is the set of clusters from \mathcal{P}_n that still exist in \mathcal{P}_m . By our definition of Y, Z , and ℓ , we conclude that any cluster $A \in \mathcal{P}_m \cap \mathcal{P}_n$ is (Z, opt_k)-connected to another cluster $B \in \mathcal{P}_m$. If we assume that there are no two (Z, opt_k)-connected clusters in $\mathcal{P}_m \cap \mathcal{P}_n$, this implies $B \in \mathcal{P}_m \setminus \mathcal{P}_n$ (see Fig. 5). Thus, using $A \in \mathcal{P}_n$, $B \in \mathcal{P}_m$, and Inequality (8), the radius of $A \cup B$ can be bounded by

$$\text{rad}(A \cup B) \leq \text{cost}_{\text{rad}}(\mathcal{P}_m) + 2 \cdot \text{cost}_{\text{rad}}(\mathcal{P}_n) + 2 \text{opt}_k. \quad (9)$$

Moreover, using a similar argument, we derive the same bound for two clusters $A_1, A_2 \in \mathcal{P}_m \cap \mathcal{P}_n$ that are (Z, opt_k)-connected to the same cluster $B \in \mathcal{P}_m \setminus \mathcal{P}_n$. That is,

$$\text{rad}(A_1 \cup A_2) \leq \text{cost}_{\text{rad}}(\mathcal{P}_m) + 2 \cdot \text{cost}_{\text{rad}}(\mathcal{P}_n) + 2 \text{opt}_k. \quad (10)$$

Next, we show that there exist at least $\lceil \frac{|\mathcal{P}_m \cap \mathcal{P}_n|}{2} \rceil$ disjoint pairs of clusters from \mathcal{P}_m such that the radius of their union can be bounded either by Inequality (9) or by Inequality (10). To do so, we first consider the pairs of clusters from $\mathcal{P}_m \cap \mathcal{P}_n$ that are (Z, opt_k)-connected to the same cluster from $\mathcal{P}_m \setminus \mathcal{P}_n$ until no candidates are left. For these pairs, we can bound the radius of their union by Inequality (10). Then, each cluster from $\mathcal{P}_m \setminus \mathcal{P}_n$ is (Z, opt_k)-connected to at most one of the remaining clusters from $\mathcal{P}_m \cap \mathcal{P}_n$. Thus, each remaining cluster $A \in \mathcal{P}_m \cap \mathcal{P}_n$ can be paired with a different cluster $B \in \mathcal{P}_m \setminus \mathcal{P}_n$ such that A and B are (Z, opt_k)-connected. For these pairs, we can bound the radius of their union by Inequality (9). Since for all pairs either one or both of the clusters come from the set $\mathcal{P}_m \cap \mathcal{P}_n$, we can lower bound the number of pairs by $\lceil \frac{|\mathcal{P}_m \cap \mathcal{P}_n|}{2} \rceil$.

To complete the proof, we show that $m - \ell \leq \lceil \frac{|\mathcal{P}_m \cap \mathcal{P}_n|}{2} \rceil$. In each iteration of its loop, the algorithm can merge at most two clusters from \mathcal{P}_n . Therefore, there are

at least $\lceil \frac{n - |\mathcal{P}_m \cap \mathcal{P}_n|}{2} \rceil$ merge steps between the computations of \mathcal{P}_n and \mathcal{P}_m . Hence, $m \leq n - \lceil \frac{n - |\mathcal{P}_m \cap \mathcal{P}_n|}{2} \rceil \leq \frac{n}{2} + \frac{|\mathcal{P}_m \cap \mathcal{P}_n|}{2}$. Using $n \leq 2\ell$, we get $m - \ell \leq \frac{|\mathcal{P}_m \cap \mathcal{P}_n|}{2}$. \square

Lemma 18 *Let $n \in \mathbb{N}$ with $n \leq 2\ell$ and $\ell < n \leq |Y|$. Then,*

$$\text{cost}_{\text{rad}}(\mathcal{P}_{\ell+1}) < 2(\log_2(\ell) + 2) \cdot (\text{cost}_{\text{rad}}(\mathcal{P}_n) + \text{opt}_k).$$

Proof For $n = \ell + 1$ there is nothing to show. Hence, assume $n > \ell + 1$. Then, by definition of Z , there exist two (Z, opt_k) -connected clusters in \mathcal{P}_n . Now let $\tilde{n} \in \mathbb{N}$ with $\tilde{n} < n$ be maximal such that no two (Z, opt_k) -connected clusters exist in $\mathcal{P}_{\tilde{n}} \cap \mathcal{P}_n$. The number \tilde{n} is well-defined since $|\mathcal{P}_1| = 1$ implies $\tilde{n} \geq 1$. It follows that the same holds for all $m \in \mathbb{N}$ with $m \leq \tilde{n}$. We conclude that Lemma 17 is applicable for all $m \in \mathbb{N}$ with $\ell < m \leq \tilde{n}$.

By the definition of \tilde{n} there still exist at least two (Z, opt_k) -connected clusters in $\mathcal{P}_{\tilde{n}+1} \cap \mathcal{P}_n$. Then, Observation 11 implies

$$\text{cost}_{\text{rad}}(\mathcal{P}_{\tilde{n}}) \leq 2 \cdot \text{cost}_{\text{rad}}(\mathcal{P}_n) + \text{opt}_k. \quad (11)$$

If $\tilde{n} \leq \ell + 1$ then Inequality (11) proves the lemma. For $\tilde{n} > \ell + 1$ let $u := \lceil \log_2(\tilde{n} - \ell) \rceil$ and define $m_i := \lceil (\frac{1}{2})^i (\tilde{n} - \ell) + \ell \rceil > \ell$ for all $i = 0, \dots, u$. Then, $m_0 = \tilde{n}$ and $m_u = \ell + 1$. Furthermore, we obtain

$$\begin{aligned} \left\lfloor \frac{m_i + \ell}{2} \right\rfloor &= \left\lfloor \frac{1}{2} \left\lceil \left(\frac{1}{2}\right)^i (\tilde{n} - \ell) + \ell \right\rceil + \frac{\ell}{2} \right\rfloor \leq \left\lfloor \frac{1}{2} \left(\left(\frac{1}{2}\right)^i (\tilde{n} - \ell) + \ell + 1 \right) + \frac{\ell}{2} \right\rfloor \\ &= \left\lfloor \left(\frac{1}{2}\right)^{i+1} (\tilde{n} - \ell) + \ell + \frac{1}{2} \right\rfloor \leq \left\lceil \left(\frac{1}{2}\right)^{i+1} (\tilde{n} - \ell) + \ell \right\rceil = m_{i+1}. \end{aligned}$$

Since Algorithm 2 uses a greedy strategy, we deduce $\text{cost}_{\text{rad}}(\mathcal{P}_{m_{i+1}}) \leq \text{cost}_{\text{rad}}(\mathcal{P}_{\lfloor \frac{m_i + \ell}{2} \rfloor})$ for all $i = 0, \dots, u - 1$. Combining this with Lemma 17 (applied to $m = m_i$), we obtain

$$\text{cost}_{\text{rad}}(\mathcal{P}_{m_{i+1}}) \leq \text{cost}_{\text{rad}}(\mathcal{P}_{m_i}) + 2 \cdot \text{cost}_{\text{rad}}(\mathcal{P}_n) + 2 \text{opt}_k.$$

By repeatedly applying this inequality for $i = 0, \dots, u - 1$ and summing up the costs, we get

$$\begin{aligned} \text{cost}_{\text{rad}}(\mathcal{P}_{m_u}) &< \text{cost}_{\text{rad}}(\mathcal{P}_{\tilde{n}}) + 2u \cdot (\text{cost}_{\text{rad}}(\mathcal{P}_n) + \text{opt}_k) \\ &\stackrel{(11)}{<} 2(u + 1) \cdot (\text{cost}_{\text{rad}}(\mathcal{P}_n) + \text{opt}_k). \end{aligned}$$

Since $\tilde{n} < 2\ell$, we get $u < \log_2(\ell) + 1$ and the lemma follows using $m_u = \ell + 1$. \square

The following lemma finishes the analysis except for the last merge step.

Lemma 19 *Let $Y \subset \mathbb{R}^d$ be finite and $\ell \leq |Y|$ such that Y is (ℓ, opt_k) -coverable. Furthermore, let $Z \subset \mathbb{R}^d$ with $|Z| = \ell$ such that for all $n \in \mathbb{N}$ with $\ell + 1 \leq n \leq |Y|$ every*

AGGLOMERATIVECOMPLETELINKAGE(X):

X finite set of input points from \mathbb{R}^d

```

1:  $\mathcal{C}_{|X|} := \{\{x\} \mid x \in X\}$ 
2: for  $i = |X| - 1, \dots, 1$  do
3:   find distinct clusters  $A, B \in \mathcal{C}_{i+1}$  minimizing  $\text{diam}(A \cup B)$ 
4:    $\mathcal{C}_i := (\mathcal{C}_{i+1} \setminus \{A, B\}) \cup \{A \cup B\}$ 
5: end for
6: return  $\mathcal{C}_1, \dots, \mathcal{C}_{|X|}$ 

```

Algorithm 3: The agglomerative complete linkage clustering algorithm

cluster in \mathcal{P}_n is (Z, opt_k) -connected to another cluster in \mathcal{P}_n , where $\mathcal{P}_1, \dots, \mathcal{P}_{|Y|}$ denotes the hierarchical clustering computed by Algorithm 2 on input Y . Then,

$$\text{cost}_{\text{rad}}(\mathcal{P}_{\ell+1}) < 2(\log_2(\ell) + 2) \cdot (24d \cdot e^{24d} + 1) \cdot \text{opt}_k.$$

Proof Let $n := \min(|Y|, 2\ell)$. Then, using Proposition 13, we get $\text{cost}_{\text{rad}}(\mathcal{P}_n) < 24d \cdot e^{24d} \cdot \text{opt}_k$. The lemma follows by using this bound in combination with Lemma 18. \square

3.2.4 Proof of Theorem 12

Using Lemma 16, we know that there is a subset $Y \subseteq X$, a number $\ell \leq k$, and a hierarchical clustering $\mathcal{P}_1, \dots, \mathcal{P}_{|Y|}$ of Y with $\text{cost}_{\text{rad}}(\mathcal{C}_k) \leq \text{cost}_{\text{rad}}(\mathcal{P}_\ell)$. Furthermore, there is a set $Z \subset \mathbb{R}^d$ such that every cluster from $\mathcal{P}_{\ell+1}$ is (Z, opt_k) -connected to another cluster in $\mathcal{P}_{\ell+1}$. Thus, $\mathcal{P}_{\ell+1}$ contains two clusters A, B that intersect with the same ball of radius opt_k . Hence

$$\text{cost}_{\text{rad}}(\mathcal{C}_k) \leq \text{rad}(A \cup B) \leq 2 \cdot \text{cost}_{\text{rad}}(\mathcal{P}_{\ell+1}) + \text{opt}_k.$$

The theorem follows using Lemma 19 and $\ell \leq k$.

3.3 Diameter k -Clustering

In this section, we analyze the agglomerative complete linkage clustering algorithm for Problem 3 stated as Algorithm 3. Again, the only difference to Algorithms 1 and 2 is the minimization of the diameter in Step 3. As in the analysis of Algorithm 2, we may assume that for any input set X the clusterings computed by Algorithm 3 are uniquely determined, i.e. the minimum in Step 3 is always unambiguous.

Note that in this section *cost* always means diameter cost and opt_k refers to the cost of an optimal diameter k -clustering of $X \subset \mathbb{R}^d$ where $k \in \mathbb{N}$ with $k \leq |X|$. Analogously to the (discrete) radius case, any cluster C is contained in a ball of radius $\text{diam}(C)$ and thus the set X is (k, opt_k) -coverable.

Observation 20 (Analogous to Observations 7 and 11) *The cost of all computed clusterings is equal to the diameter of the cluster created last. Furthermore, the diameter of the union of any two clusters is always an upper bound for the cost of the clustering to be computed next.*

The following theorem states our main result.

Theorem 21 *Let $X \subset \mathbb{R}^d$ be a finite set of points. Then, for all $k \in \mathbb{N}$ with $k \leq |X|$, the partition \mathcal{C}_k of X into k clusters as computed by Algorithm 3 satisfies*

$$\text{cost}_{\text{diam}}(\mathcal{C}_k) = O(\log k) \cdot \text{opt}_k,$$

where opt_k denotes the cost of an optimal solution to Problem 3, and the constant hidden in the O -notation is doubly exponential in the dimension d .

As in the proof of Theorems 6 and 12, we first show a bound for the cost of the intermediate $2k$ -clustering. However, we have to apply a different analysis again. This time, the new analysis results in a bound that depends doubly exponential on the dimension.

3.3.1 Analysis of the $2k$ -Clustering

Proposition 22 *Let $X \subset \mathbb{R}^d$ be finite. Then, for all $k \in \mathbb{N}$ with $2k \leq |X|$, the partition \mathcal{C}_{2k} of X into $2k$ clusters as computed by Algorithm 3 satisfies*

$$\text{cost}_{\text{diam}}(\mathcal{C}_{2k}) < 2^{3\sigma} (28d + 6) \cdot \text{opt}_k,$$

where $\sigma = (42d)^d$ and opt_k denotes the cost of an optimal solution to Problem 3.

In our analysis of the k -center problem, we made use of the fact that merging two clusters lying inside a ball of some radius r results in a new cluster of radius at most r . This is no longer true for the diameter k -clustering problem. We are not able to derive a bound for the diameter of the new cluster that is significantly less than $2r$. The additional factor of 2 makes our analysis from Sect. 3.2.1 useless for the diameter case.

To prove Proposition 22, we divide the merge steps of Algorithm 3 into two stages. The first stage consists of the merge steps down to a $2^{2^{O(d \log d)}}$ k -clustering. The analysis of the first stage is based on the following notion of similarity. Two clusters are called similar if one cluster can be translated such that every point of the translated cluster is near a point of the second cluster. Then, by merging similar clusters, the diameter essentially increases by the length of the translation vector. During the first stage, we guarantee that there is a sufficiently large number of similar clusters left. The cost of the intermediate $2^{2^{O(d \log d)}}$ k -clustering can be upper bounded by $O(d) \cdot \text{opt}_k$.

The second stage consists of the steps reducing the number of remaining clusters from $2^{2^{O(d \log d)}}$ k to only $2k$. In this stage, we are no longer able to guarantee that a sufficiently large number of similar clusters exists. Therefore, we analyze the merge steps of the second stage using a weaker argument, very similar to the one used in the second step of the analysis in the discrete k -center case (cf. Sect. 3.1.2). As long as there are more than $2k$ clusters left, we are able to find sufficiently many pairs of clusters that intersect with the same cluster of an optimal k -clustering. Therefore, we can bound the cost of merging such a pair by the sum of the diameters of the

two clusters plus the diameter of the optimal cluster. We find that the cost of the intermediate $2k$ -clustering is upper bounded by $2^{2^{O(d \log d)}} \cdot \text{opt}_k$. Let us remark that we do not obtain our main result if we already use this argument for the first stage.

Both stages are again subdivided into phases, such that in each phase the number of remaining clusters is reduced by one fourth.

3.3.2 Stage One

The following lemma will be used to bound the increase of the cost during a single phase.

Lemma 23 *Let $\lambda \in \mathbb{R}$ with $0 < \lambda < 1$ and $\rho := \lceil (\frac{3}{\lambda})^d \rceil$. Furthermore, let $m \in \mathbb{N}$ with $2^{\rho+1}k < m \leq |X|$. Then,*

$$\text{cost}_{\text{diam}}(\mathcal{C}_{\lfloor \frac{3m}{4} \rfloor}) < (1 + 2\lambda) \cdot \text{cost}_{\text{diam}}(\mathcal{C}_m) + 4\sqrt{\frac{2^{\rho+1}k}{m}} \cdot \text{opt}_k. \quad (12)$$

Proof From every cluster $C \in \mathcal{C}_m$, we fix an arbitrary point and denote it by p_C . Let $R := \text{cost}_{\text{diam}}(\mathcal{C}_m)$. Then, the distance from p_C to any $q \in C$ is at most R and we get $C - p_C \subset B_R^d(0)$.

A ball of radius R can be covered by ρ balls of radius λR (see [13]). Hence, there exist $y_1, \dots, y_\rho \in \mathbb{R}^d$ with $B_R^d(0) \subseteq \bigcup_{i=1}^\rho B_{\lambda R}^d(y_i)$. For $C \in \mathcal{C}_m$, we call the set $\text{Conf}(C) := \{y_i \mid 1 \leq i \leq \rho \text{ and } B_{\lambda R}^d(y_i) \cap (C - p_C) \neq \emptyset\}$ the configuration of C . That is, we identify each cluster $C \in \mathcal{C}_m$ with the subset of the balls $B_{\lambda R}^d(y_1), \dots, B_{\lambda R}^d(y_\rho)$ that intersect with $C - p_C$. Note that no cluster from $C \in \mathcal{C}_m$ has an empty configuration. The number of possible configurations is upper bounded by 2^ρ .

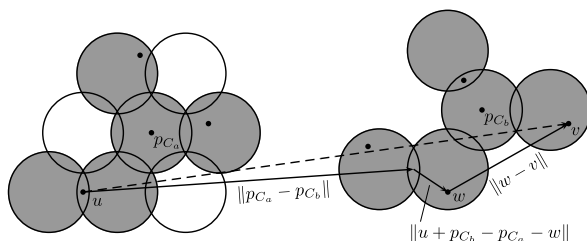
Let $t := \lfloor \frac{3m}{4} \rfloor$ and $\mathcal{C}_m \cap \mathcal{C}_{t+1}$ be the set of clusters from \mathcal{C}_m that still exist $\lfloor \frac{m}{4} \rfloor - 1$ merge steps after the computation of \mathcal{C}_m . In each iteration of its loop, the algorithm can merge at most two clusters from \mathcal{C}_m . Thus, $|\mathcal{C}_m \cap \mathcal{C}_{t+1}| > \frac{m}{2}$. It follows that there exist $j > \frac{m}{2^{\rho+1}}$ distinct clusters $C_1, \dots, C_j \in \mathcal{C}_m \cap \mathcal{C}_{t+1}$ with the same configuration. Using $m > 2^{\rho+1}k$, we deduce $j > k$.

Let $P := \{p_{C_1}, \dots, p_{C_j}\}$. Since X is (k, opt_k) -coverable, so is $P \subset X$. Therefore, by Lemma 5, there exist distinct $a, b \in \{1, \dots, j\}$ such that $\|p_{C_a} - p_{C_b}\| \leq 4\sqrt{\frac{2^{\rho+1}k}{m}} \cdot \text{opt}_k$.

Next, we want to bound the diameter of the union of the corresponding clusters C_a and C_b . The distance between any two points $u, v \in C_a$ or $u, v \in C_b$ is at most the cost of \mathcal{C}_m . Now let $u \in C_a$ and $v \in C_b$. Using the triangle inequality, for any $w \in \mathbb{R}^d$, we obtain $\|u - v\| \leq \|p_{C_a} - p_{C_b}\| + \|u + p_{C_b} - p_{C_a} - w\| + \|w - v\|$ (see Fig. 6).

For $\|p_{C_a} - p_{C_b}\|$, we just derived an upper bound. To bound $\|u + p_{C_b} - p_{C_a} - w\|$, we let $y \in \text{Conf}(C_a) = \text{Conf}(C_b)$ such that $u - p_{C_a} \in B_{\lambda R}^d(y)$. Furthermore, we fix $w \in C_b$ with $w - p_{C_b} \in B_{\lambda R}^d(y)$. Hence, $\|u + p_{C_b} - p_{C_a} - w\| = \|u - p_{C_a} - (w - p_{C_b})\|$ can be upper bounded by $2\lambda R = 2\lambda \cdot \text{cost}_{\text{diam}}(\mathcal{C}_m)$. For $w \in C_b$ the distance $\|w - v\|$ is bounded by $\text{diam}(C_b) \leq \text{cost}_{\text{diam}}(\mathcal{C}_m)$. We conclude that merging clusters

Fig. 6 Congruent configurations



C_a and C_b results in a cluster whose diameter can be upper bounded by

$$\text{diam}(C_a \cup C_b) < (1 + 2\lambda) \cdot \text{cost}_{\text{diam}}(C_m) + 4\sqrt[d]{\frac{2^{\rho+1}k}{m}} \cdot \text{opt}_k.$$

Using Observation 20 and the fact that C_a and C_b are part of the clustering \mathcal{C}_{t+1} , we can upper bound the cost of \mathcal{C}_t by $\text{cost}_{\text{diam}}(\mathcal{C}_t) \leq \text{diam}(C_a \cup C_b)$. \square

Note that the parameter λ from Lemma 23 establishes a trade-off between the two terms on the right-hand side of Inequality (12). To complete the analysis of the first stage, we have to carefully choose λ . In the proof of the following lemma, we use $\lambda = \frac{\ln \frac{4}{3}}{4d}$ and apply Lemma 23 for $\lceil \log_{\frac{4}{3}} \frac{|X|}{2^{\sigma+1}k} \rceil$ consecutive phases, where $\sigma = (42d)^d$. Then, we are able to upper bound the total increase of the cost by a term that is linear in d and r and independent of $|X|$ and k . The number of remaining clusters is independent of the number of input points $|X|$ and only depends on the dimension d and the desired number of clusters k .

Lemma 24 *Let $2^{\sigma+1}k < |X|$ for $\sigma = (42d)^d$. Then, on input X , Algorithm 3 computes a clustering $\mathcal{C}_{2^{\sigma+1}k}$ with $\text{cost}_{\text{diam}}(\mathcal{C}_{2^{\sigma+1}k}) < (28d + 4) \cdot \text{opt}_k$.*

Proof Let $u := \lceil \log_{\frac{4}{3}} \frac{2^{\sigma+1}k}{|X|} \rceil$ and define $m_i := \lceil (\frac{3}{4})^i |X| \rceil$ for all $i = 0, \dots, u$. Furthermore, let $\lambda = \frac{\ln \frac{4}{3}}{4d}$. This implies $\rho \leq \sigma$ for the parameter ρ of Lemma 23. Then, $m_u \leq 2^{\sigma+1}k$ and $m_i > 2^{\sigma+1}k \geq 2^{\rho+1}k$ for all $i = 0, \dots, u - 1$. Since $\lfloor \frac{3m_i}{4} \rfloor = \lfloor \frac{3}{4} \lceil (\frac{3}{4})^i |X| \rceil \rfloor \leq \lfloor (\frac{3}{4})^{i+1} |X| + \frac{3}{4} \rfloor \leq \lceil (\frac{3}{4})^{i+1} |X| \rceil = m_{i+1}$ and Algorithm 3 uses a greedy strategy, we deduce $\text{cost}_{\text{diam}}(\mathcal{C}_{m_{i+1}}) \leq \text{cost}_{\text{diam}}(\mathcal{C}_{\lfloor \frac{3m_i}{4} \rfloor})$ for all $i = 0, \dots, u - 1$. Combining this with Lemma 23 (applied to $m = m_i$), we obtain

$$\text{cost}_{\text{diam}}(\mathcal{C}_{m_{i+1}}) < (1 + 2\lambda) \cdot \text{cost}_{\text{diam}}(\mathcal{C}_{m_i}) + 4\sqrt[d]{\frac{2^{\rho+1}k}{m_i}} \cdot \text{opt}_k.$$

By repeatedly applying this inequality for $i = 0, \dots, u - 1$ and using

$$\text{cost}_{\text{diam}}(\mathcal{C}_{2^{\sigma+1}k}) \leq \text{cost}_{\text{diam}}(\mathcal{C}_{m_u}) \quad \text{and} \quad \text{cost}_{\text{diam}}(\mathcal{C}_{m_0}) = 0,$$

we get

$$\begin{aligned} \text{cost}_{\text{diam}}(C_{2^{\sigma+1}k}) &< \sum_{i=0}^{u-1} \left((1+2\lambda)^i \cdot 4 \sqrt[d]{\frac{2^{\sigma+1}k}{(\frac{3}{4})^{u-1-i}|X|}} \text{opt}_k \right) \\ &= 4 \sqrt[d]{\frac{2^{\sigma+1}k}{(\frac{3}{4})^{u-1}|X|}} \text{opt}_k \cdot \sum_{i=0}^{u-1} \left((1+2\lambda)^i \cdot \sqrt[d]{\left(\frac{3}{4}\right)^i} \right). \end{aligned}$$

Using $u-1 < \log_{\frac{3}{4}} \frac{2^{\sigma+1}k}{|X|}$, we deduce

$$\text{cost}_{\text{diam}}(C_{2^{\sigma+1}k}) < 4 \text{opt}_k \cdot \sum_{i=0}^{u-1} \left(\frac{1+2\lambda}{\sqrt[d]{\frac{4}{3}}} \right)^i. \quad (13)$$

By taking only the first two terms of the series expansion of the exponential function, we get $1+2\lambda = 1 + \frac{\ln \frac{4}{3}}{2d} < e^{\frac{\ln \frac{4}{3}}{2d}} = \sqrt[d]{\frac{4}{3}}$. Substituting this bound into Inequality (13) and extending the sum gives

$$\text{cost}_{\text{diam}}(C_{2^{\sigma+1}k}) < 4 \text{opt}_k \cdot \sum_{i=0}^{\infty} \left(\frac{1}{\sqrt[d]{\frac{4}{3}}} \right)^i < 4 \text{opt}_k \cdot \sum_{i=0}^{\infty} \left(\frac{1}{1+2\lambda} \right)^i.$$

Solving the geometric series leads to

$$\text{cost}_{\text{diam}}(C_{2^{\sigma+1}k}) < 4 \left(\frac{1}{2\lambda} + 1 \right) \cdot \text{opt}_k < (28d+4) \cdot \text{opt}_k. \quad \square$$

3.3.3 Stage Two

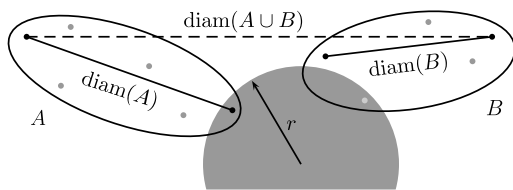
The second stage covers the remaining merge steps until Algorithm 3 computes the clustering C_{2k} . However, compared to stage one, the analysis of a single phase yields a weaker bound. The following lemma provides an analysis of a single phase of the second stage. It is very similar to Lemmas 9 and 10 in the analysis of the discrete k -center problem.

Lemma 25 *Let $m \in \mathbb{N}$ with $2k < m \leq |X|$. Then,*

$$\text{cost}_{\text{diam}}(C_{\lfloor \frac{3m}{4} \rfloor}) < 2 \cdot (\text{cost}_{\text{diam}}(C_m) + \text{opt}_k).$$

Proof Let $t := \lfloor \frac{3m}{4} \rfloor$. Then, $C_m \cap C_{t+1}$ is the set of clusters from C_m which still exist $\lceil \frac{m}{4} \rceil - 1 < \frac{m}{4}$ merge steps after the computation of C_m . In each iteration of its loop the algorithm can merge at most two clusters from C_m . Thus, $|C_m \cap C_{t+1}| > \frac{m}{2} > k$. Since X is (k, opt_k) -coverable there exists a point $y \in \mathbb{R}^d$ such that $B_{\text{opt}_k}^d(y)$ intersects with two clusters $A, B \in C_m \cap C_{t+1}$. We conclude that merging A and B would result in a cluster whose diameter can be upper bounded by $\text{diam}(A \cup B) < 2 \text{cost}_{\text{diam}}(C_m) + 2 \text{opt}_k$ (cf. Fig. 7). The result follows using $A, B \in C_{t+1}$ and Observation 20. \square

Fig. 7 Merging two clusters intersecting with a ball of radius r



Lemma 26 Let $n \in \mathbb{N}$ with $n \leq 2^{\sigma+1}k$ and $2k < n \leq |X|$ for $\sigma = (42d)^d$. Then, on input X , Algorithm 3 computes a clustering \mathcal{C}_{2k} with

$$\text{cost}_{\text{diam}}(\mathcal{C}_{2k}) < 2^{3\sigma} (\text{cost}_{\text{diam}}(\mathcal{C}_n) + 2 \text{opt}_k).$$

Proof Let $u := \lceil \log_{\frac{3}{4}} \frac{2k}{n} \rceil$ and define $m_i := \lceil (\frac{3}{4})^i n \rceil$ for all $i = 0, \dots, u$. Then, $m_u \leq 2k$ and $m_i > 2k$ for all $i = 0, \dots, u-1$. Analogously to the proof of Lemma 24, we get $\lfloor \frac{3m_i}{4} \rfloor \leq m_{i+1}$ and using Lemma 25, we deduce $\text{cost}_{\text{diam}}(\mathcal{C}_{m_{i+1}}) < 2 \cdot (\text{cost}_{\text{diam}}(\mathcal{C}_{m_i}) + \text{opt}_k)$ for all $i = 0, \dots, u-1$. By repeatedly applying this inequality and using $\text{cost}_{\text{diam}}(\mathcal{C}_{2k}) \leq \text{cost}_{\text{diam}}(\mathcal{C}_{m_u})$, we get $\text{cost}_{\text{diam}}(\mathcal{C}_{2k}) < 2^u \cdot (\text{cost}_{\text{diam}}(\mathcal{C}_n) + 2 \text{opt}_k)$. Hence using $u \leq \lceil \log_{\frac{3}{4}} 2^{\sigma} \rceil < 3\sigma$, the result follows. \square

Proposition 22 follows immediately by combining Lemmas 24 and 26.

3.3.4 Analysis of the Remaining Merge Steps

We analyze the remaining merge steps analogously to the k -center problem. Therefore, in this section we only discuss the differences, most of which are slightly modified bounds for the cost of merging two clusters (cf. Fig. 7).

The connectivity property from Sect. 3.2.2 remains the same. However, for any two (Z, r) -connected clusters A, B , we use

$$\text{diam}(A \cup B) \leq \text{diam}(A) + \text{diam}(B) + 2r \quad (14)$$

as a replacement for Inequality (8). Furthermore, Lemma 16 also holds for the diameter k -clustering problem, i.e. with $\text{cost}_{\text{diam}}(\mathcal{C}_k) \leq \text{cost}_{\text{diam}}(\mathcal{P}_\ell)$.

Using Inequality (14) in the proof of Lemma 17, we get

$$\text{diam}(A \cup B) \leq \text{cost}_{\text{diam}}(\mathcal{P}_m) + \text{cost}_{\text{diam}}(\mathcal{P}_n) + 2 \text{opt}_k$$

as a replacement for Inequality (9) while Inequality (10) can be replaced by

$$\text{diam}(A_1 \cup A_2) \leq \text{cost}_{\text{diam}}(\mathcal{P}_m) + 2 \cdot (\text{cost}_{\text{diam}}(\mathcal{P}_n) + 2 \text{opt}_k).$$

That is, for the diameter k -clustering problem the two upper bounds are different. However, the second one is larger than the first one. Using it in both cases, the inequality from Lemma 17 changes slightly to

$$\text{cost}_{\text{diam}}(\mathcal{P}_{\lfloor \frac{m+\ell}{2} \rfloor}) \leq \text{cost}_{\text{diam}}(\mathcal{P}_m) + 2 \cdot (\text{cost}_{\text{diam}}(\mathcal{P}_n) + 2 \text{opt}_k).$$

Together with $\text{cost}_{\text{diam}}(\mathcal{P}_{\tilde{n}}) \leq 2 \cdot \text{cost}_{\text{diam}}(\mathcal{P}_n) + 2 \text{opt}_k$ as a replacement for Inequality (11), the bound from Lemma 18 becomes

$$\text{cost}_{\text{diam}}(\mathcal{P}_{\ell+1}) < 2(\log_2(\ell) + 2) \cdot (\text{cost}_{\text{diam}}(\mathcal{P}_n) + 2 \text{opt}_k).$$

Thus, using Proposition 22, the upper bound for the cost of the $\ell + 1$ -clustering of Y from Lemma 19 becomes

$$\text{cost}_{\text{diam}}(\mathcal{P}_{\ell+1}) < 2(\log_2(\ell) + 2) \cdot (2^{3\sigma}(28d + 6) + 2) \cdot \text{opt}_k$$

for $\sigma = (42d)^d$. Analogously to Sect. 3.2.4, this proves Theorem 21.

3.4 Analysis of the One-Dimensional Case

For $d = 1$, we are able to show that Algorithm 3 computes an approximation to Problem 3 with an approximation factor of at most 3. We even know that for any input set $X \subset \mathbb{R}$ the approximation factor of the computed solution is strictly below 3. However, we do not show an approximation factor of $3 - \epsilon$ for some $\epsilon > 0$. The proof of this upper bound is very technical, makes extensive use of the total order of the real numbers, and is certainly not generalizable to higher dimensions. Therefore, we omit it.

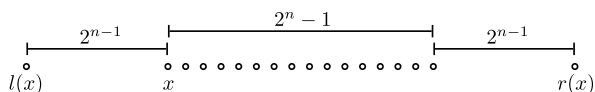
4 Lower Bounds

In this section, we present constructions of several input sets yielding lower bounds for the approximation factor of Algorithm 3. To this end, we look into possible runs of the algorithm. Whenever Algorithm 3 is able to choose between several possible merge steps generating a cluster of equal minimum diameter, we simply assume that we can govern its choice.

In Sect. 4.1, we show that for any input set $X \subset \mathbb{R}$ (i.e. $d = 1$) Algorithm 3 has an approximation factor of at least 2.5. In Sect. 3.4, we stated that in this case Algorithm 3 computes a solution to Problem 3 with approximation factor strictly below 3. Hence, for $d = 1$, we obtain almost matching upper and lower bounds for the cost of the solution computed by Algorithm 3.

Furthermore, in Sect. 4.2, we show that the dimension d has an impact on the approximation factor of Algorithm 3. This follows from a 2-dimensional input set yielding a lower bound of 3 for the metric based on the ℓ_∞ -norm. Note that this exceeds the upper bound from the one-dimensional case.

Moreover, in Sect. 4.4, we show that there exist input instances such that Algorithm 3 computes an approximation to Problem 3 with an approximation factor of $\Omega(\sqrt[p]{\log k})$ for metrics based on an ℓ_p -norm ($1 \leq p < \infty$) and $\Omega(\log k)$ for the metric based on the ℓ_∞ -norm. In case of the ℓ_1 - and the ℓ_∞ -norm, this matches the already known lower bound [4] that has been shown using a rather artificial metric. However, the bound in [4] is derived from a 2-dimensional input set, while in our instances the dimension depends on k .

Fig. 8 A sketch of the set $W(x)$ 

Finally, we will see that the lower bound of $\Omega(\sqrt[n]{\log k})$ for any ℓ_p -norm and $\Omega(\log k)$ for the ℓ_∞ -norm can be adapted to the discrete k -center problem (see Sect. 4.4.1). In case of the ℓ_2 -norm, we thus obtain almost matching upper and lower bounds for the cost of the solution computed by Algorithm 1. Furthermore, we will be able to restrict the dependency on d and k of the approximation factor of Algorithm 1.

4.1 Any Metric and $d = 1$

We first show a lower bound for the approximation factor of Algorithm 3 using a sequence of input sets from \mathbb{R}^d with $d = 1$. Since up to normalization there is only one metric for $d = 1$, without loss of generality we assume the Euclidean metric.

Proposition 27 *For all $\varepsilon > 0$ and $k \geq 4$ there exists an input set $X \subset \mathbb{R}$ such that Algorithm 3 computes a solution to Problem 3 with cost at least $\frac{5}{2} - \varepsilon$ times the cost of an optimal solution.*

Proof We show how to construct an input set for $k = 4$. The construction can easily be extended for $k > 4$. For any fixed $n \in \mathbb{N}$, we consider the following instance. For $x \in \mathbb{R}$, we define a set $V(x)$ consisting of 2^n equidistant points:

$$V(x) := \{x + i \mid i \in \mathbb{N} \text{ and } 0 \leq i < 2^n\}.$$

That is, neighboring points are at distance 1 and $\text{diam}(V(x)) = 2^n - 1$. Furthermore, we define:

$$\begin{aligned} l(x) &:= x - 2^{n-1}, \\ r(x) &:= x + 2^n - 1 + 2^{n-1} = x + 3 \cdot 2^{n-1} - 1, \\ W(x) &:= V(x) \cup \{l(x), r(x)\}. \end{aligned}$$

It follows that $\text{diam}(W(x)) = 2^{n+1} - 1$ as shown in Fig. 8.

We define the following input set X :

$$X := \bigcup_{i=1}^4 W(x_i)$$

where $x_i := i \cdot (7 \cdot 2^{n-1} - 2)$ for $i = 1, \dots, 4$. Then, there is a gap of $3 \cdot 2^{n-1} - 1$ between $W_n(x_i)$ and $W_n(x_{i+1})$, i.e.

$$\text{diam}(\{r(x_i), l(x_{i+1})\}) = 3 \cdot 2^{n-1} - 1 \quad \text{for } i = 1, \dots, 3. \quad (15)$$

The optimal 4-clustering of X is

$$\mathcal{C}_4^{\text{opt}} := \{W(x_1), W(x_2), W(x_3), W(x_4)\}$$

Fig. 9 A part of the dendrogram for $W(x_i)$

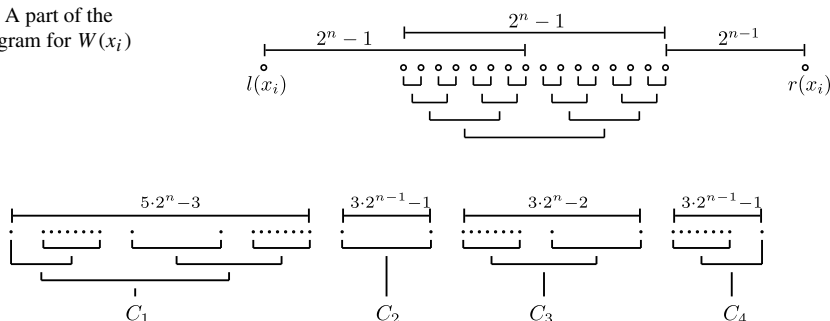


Fig. 10 A part of the dendrogram for X

and $\text{cost}_{\text{diam}}(C_4^{\text{opt}}) = 2^{n+1} - 1$. However, the solution computed by Algorithm 3 may be worse. At the beginning, the minimum distance between two points from X is 1. The possible pairs of points with distance 1 come from the sets $V(x_i)$ for $i = 1, \dots, 4$. Since the distance between $V(x_i)$ and $l(x_i)$ or $r(x_i)$ is 2^{n-1} , we can assume that the algorithm merges all points of $V(x_i)$ for $i = 1, \dots, 4$ as shown in Fig. 9. It follows that Algorithm 3 computes the following 12-clustering:

$$\begin{aligned} \mathcal{C}_{12} = & \{ \{l(x_1)\}, V(x_1), \{r(x_1)\}, \\ & \{l(x_2)\}, V(x_2), \{r(x_2)\}, \\ & \{l(x_3)\}, V(x_3), \{r(x_3)\}, \\ & \{l(x_4)\}, V(x_4), \{r(x_4)\} \}. \end{aligned}$$

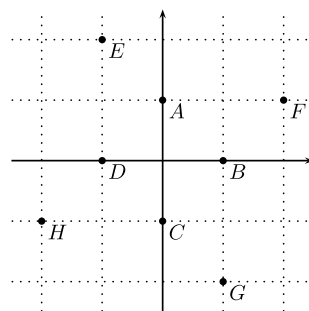
For $i = 1, \dots, 4$, the diameters of $\{l(x_i)\} \cup V(x_i)$ and $V(x_i) \cup \{r(x_i)\}$ are equal to $3 \cdot 2^{n-1} - 1$ and these are the best possible merge steps. Therefore, by (15), we can assume that Algorithm 3 merges $r(x_i)$ and $l(x_{i+1})$ for $i = 1, \dots, 3$ first. This results in the following 7-clustering:

$$\begin{aligned} \mathcal{C}_7 = & \{ \{l(x_1)\} \cup V(x_1), \\ & \{r(x_1), l(x_2)\}, V(x_2), \\ & \{r(x_2), l(x_3)\}, V(x_3), \\ & \{r(x_3), l(x_4)\}, V(x_4) \cup \{r(x_4)\} \} \end{aligned}$$

where $V(x_2)$ and $V(x_3)$ have a diameter of $2^n - 1$ while the remaining clusters have a diameter of $3 \cdot 2^{n-1} - 1$ (see Fig. 10). Between two neighboring clusters of \mathcal{C}_7 , there is a gap of 2^{n-1} .

In the next step of Algorithm 3, the best possible choice is to merge $\{r(x_1), l(x_2)\}$ with $V(x_2)$, $\{r(x_2), l(x_3)\}$ with $V(x_2)$ or $V(x_3)$, or $\{r(x_3), l(x_4)\}$ with $V(x_3)$. We let it merge $\{r(x_1), l(x_2)\}$ with $V(x_2)$ and $\{r(x_3), l(x_4)\}$ with $V(x_3)$. This results in a 5-clustering where the clusters have alternating lengths of $3 \cdot 2^{n-1} - 1$ and $3 \cdot 2^n - 2$ with gaps of 2^{n-1} between them. Then, in the step resulting in \mathcal{C}_4 , Algorithm 3 has to

Fig. 11 Lower bound for the metric based on the ℓ_∞ -norm



create a cluster of diameter $5 \cdot 2^n - 3$ as shown in Fig. 10. Therefore, the computed solution has an approximation factor of

$$\frac{\text{cost}_{\text{diam}}(C_4)}{\text{cost}_{\text{diam}}(C_4^{\text{opt}})} = \frac{5 \cdot 2^n - 3}{2^{n+1} - 1}.$$

For n going to infinity this approximation factor converges from below to $\frac{5}{2}$. \square

4.2 ℓ_∞ -Metric and $d = 2$

In this section, we give a construction that needs only eight points from \mathbb{R}^2 and yields a lower bound of 3 for the metric based on the ℓ_∞ -norm. Recall that in Sect. 3.4, we showed that for $d = 1$ the approximation factor of a computed solution is always strictly less than 3. Therefore, the lower bound of 3 for $d = 2$ implies that the dimension d has an impact on the approximation factor of Algorithm 3.

Proposition 28 *For the metric based on the ℓ_∞ -norm, there exists an input set $X \subset \mathbb{R}^2$ such that Algorithm 3 computes a solution to Problem 3 with three times the cost of an optimal solution.*

Proof We prove the proposition by constructing an example input set for $k = 4$ (see Fig. 11). Consider the following eight points in \mathbb{R}^2 :

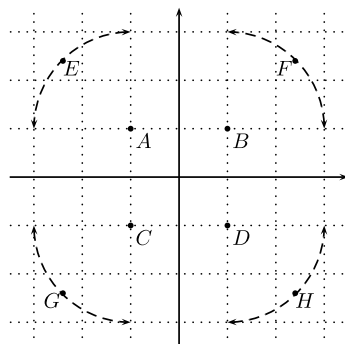
$$\begin{aligned} A &= (0, 1), & E &= (-1, 2), \\ B &= (1, 0), & F &= (2, 1), \\ C &= (0, -1), & G &= (1, -2), \\ D &= (-1, 0), & H &= (-2, -1). \end{aligned}$$

The optimal 4-clustering of these points is

$$C_4^{\text{opt}} = \{\{A, E\}, \{B, F\}, \{C, G\}, \{D, H\}\}$$

which has a maximum ℓ_∞ -diameter of 1. However, it is also possible that Algorithm 3 starts by merging A with B and C with D . Then, in the third step, the algorithm will merge E or F with $\{A, B\}$, G or H with $\{C, D\}$, or $\{A, B\}$ with $\{C, D\}$. We assume the latter. Thus, in the fourth merge step a cluster of ℓ_∞ -diameter 3 will be created. \square

Fig. 12 Lower bound for the metric based on the ℓ_2 -norm. The points C, D, G, H have a z -coordinate of 0, while the points A, B, E, F have a z -coordinate of $2\sqrt{x}$



4.3 Euclidean Metric and $d = 3$

For the Euclidean case, we are able to construct a 3-dimensional instance that yields a lower bound of 2.56. This is below the upper bound of 3 from the one-dimensional case. Therefore, this instance does not show an impact of the dimension d in the Euclidean case as in the previous section. But this lower bound is still better than the lower bound of 2.5 from the one-dimensional case. This suggests that in higher dimensions it might be easier to construct good lower bounds.

Proposition 29 *For the Euclidean metric there exists an input set $X \subset \mathbb{R}^3$ such that Algorithm 3 computes a solution to Problem 3 with cost 2.56 times the cost of an optimal solution.*

Proof We prove the proposition by constructing an example input set for $k = 4$ (see Fig. 12). For any fixed $x \in \mathbb{R}$ with $0 < x < 2$ consider the following eight points in \mathbb{R}^2 :

$$\begin{aligned} A &= (-1, 1, 2\sqrt{x}), & E &= (-(1+x), 1 + \sqrt{4-x^2}, 2\sqrt{x}), \\ B &= (1, 1, 2\sqrt{x}), & F &= (1+x, 1 + \sqrt{4-x^2}, 2\sqrt{x}), \\ C &= (-1, -1, 0), & G &= (-(1+x), -(1 + \sqrt{4-x^2}), 0), \\ D &= (1, -1, 0), & H &= (1+x, -(1 + \sqrt{4-x^2}), 0). \end{aligned}$$

The optimal 4-clustering of these points is

$$C_4^{\text{opt}} = \{\{A, E\}, \{B, F\}, \{C, G\}, \{D, H\}\},$$

which has a maximum ℓ_2 -diameter of 2. However, since $\|A - B\| = \|C - D\| = 2$ it is possible that Algorithm 3 starts by merging A with B and C with D . Then, the cheapest merge adds one of the points E, F to the cluster $\{A, B\}$ or it adds one of the points G, H to the cluster $\{C, D\}$ or it merges $\{A, B\}$ with $\{C, D\}$. We assume the latter. The resulting cluster $\{A, B, C, D\}$ has a diameter of $2\sqrt{2+x}$. Then, in the fourth merge step, the algorithm will either merge one of the pairs E, F and G, H

or one of the pairs E, G and F, H . The choice depends on the parameter x . Note that Algorithm 3 will not merge the cluster $\{A, B, C, D\}$ with one of the remaining four points, since this is always more expensive. The diameter of the created cluster is maximized for $x \approx 1.56$. If we fix $x = 1.56$, the algorithm merges E with F or G with H . This results in a 4-clustering of cost 5.12, while the optimal solution has cost 2. \square

4.4 ℓ_p -Metric ($1 \leq p \leq \infty$) in Variable Dimension

In the following, we consider the diameter k -clustering problem with respect to the metric based on the ℓ_1 -norm. We show that there exists an input instance in dimension $O(k)$ such that Algorithm 3 computes a solution with an approximation factor of $\Omega(\log k)$.

Proposition 30 *For the metric based on the ℓ_1 -norm, there exists an input set $X \subset \mathbb{R}^d$ with $d = k + \log_2 k$ such that Algorithm 3 computes a solution to Problem 3 with $\frac{1}{2} \log_2 k$ times the cost of an optimal solution.*

Proof For simplicity's sake, assume k to be a power of 2. In the sequel, we consider the $(k + \log_2 k)$ -dimensional set X of $|X| = k^2$ points defined by

$$X = \left\{ \begin{bmatrix} e_i \\ b \end{bmatrix} \mid \forall 1 \leq i \leq k \text{ and } b \in \{0, 1\}^{\log_2 k} \right\}.$$

Here, $e_i \in \mathbb{R}^k$ denotes the i -th canonical unit vector. Consider the following k -clustering

$$\mathcal{C}_k^* = \{C_b \mid b \in \{0, 1\}^{\log_2 k}\},$$

where for each $b \in \{0, 1\}^{\log_2 k}$ cluster C_b is given by

$$C_b = \left\{ \begin{bmatrix} e_i \\ b \end{bmatrix} \mid \forall 1 \leq i \leq k \right\}.$$

The largest diameter of \mathcal{C}_k^* is $\text{cost}_{\text{diam}}(\mathcal{C}_k^*) = 2$. Hence for opt_k , the diameter of an optimal solution, it holds

$$\text{opt}_k \leq 2. \quad (16)$$

However, we find that

$$\text{diam} \left(\left\{ \begin{bmatrix} e_i \\ b_1 \end{bmatrix}, \begin{bmatrix} e_j \\ b_2 \end{bmatrix} \right\} \right) = \begin{cases} h(b_1, b_2) & \text{if } i = j \\ 2 + h(b_1, b_2) & \text{if } i \neq j \end{cases}$$

where $h(b_1, b_2)$ denotes the Hamming distance between the strings $b_1, b_2 \in \{0, 1\}^{\log_2 k}$. Hence, we may assume that Algorithm 3 starts by merging points $[e_i, 0, b']^\top$ and $[e_i, 1, b']^\top$ for all $1 \leq i \leq k$ and all $b' \in \{0, 1\}^{\log_2(k)-1}$, thereby forming $\frac{1}{2}k^2$ clusters of diameter 1.

Next, we show inductively that Algorithm 3 keeps merging pairs of clusters that agree on the first k coordinates until the algorithm halts. To this end, assume that there is some number $1 \leq t \leq \log_2 k$ such that the clustering computed so far consists solely of the clusters

$$C_{i,b'}^{(t)} = \left\{ \begin{bmatrix} e_i \\ b \\ b' \end{bmatrix} \mid b \in \{0, 1\}^t \right\}$$

for all $1 \leq i \leq k$ and all $b' \in \{0, 1\}^{\log_2(k)-t}$. Also note that this is the case with $t = 1$ after the first $\frac{1}{2}k^2$ merges. In such a case, we have

$$\text{diam}(C_{i,b_1}^{(t)} \cup C_{j,b_2}^{(t)}) = \begin{cases} t + h(b_1, b_2) & \text{if } i = j \\ 2 + t + h(b_1, b_2) & \text{if } i \neq j. \end{cases}$$

Hence, as above, we may assume that in the next $\frac{1}{2^{t+1}}k^2$ steps Algorithm 3 merges the clusters $C_{i,0b'}^{(t)}$ and $C_{i,1b'}^{(t)}$ for all $1 \leq i \leq k$ and all $b' \in \{0, 1\}^{\log_2(k)-(t+1)}$. The resulting clusters are of diameter $t + 1$. Also, we have $C_{i,b'}^{(t+1)} = C_{i,0b'}^{(t)} \cup C_{i,1b'}^{(t)}$.

Algorithm 3 keeps merging clusters in this way until after $t = \log_2 k$ rounds we end up with the k -clustering $\mathcal{C}_k = \{C_i \mid 1 \leq i \leq k\}$ where

$$C_i = \left\{ \begin{bmatrix} e_i \\ b \end{bmatrix} \mid b \in \{0, 1\}^{\log_2 k} \right\}.$$

These clusters C_i are of diameter $\log_2 k$. Comparing to (16), we deduce that Algorithm 3 computes a solution to Problem 3 with at least $\frac{1}{2} \log_2 k$ times the cost of an optimal solution. \square

Considering the diameter k -clustering problem with respect to an arbitrary ℓ_p -metric (with $1 \leq p < \infty$), note that the behavior of Algorithm 3 does not change if we consider the p -th power of the ℓ_p -distance instead of the ℓ_p -distance. Also note that for all $x, y \in \{0, 1\}^d$ we have $\|x - y\|_p^p = \|x - y\|_1$. Since instance X from Proposition 30 is a subset of $\{0, 1\}^d$, we immediately obtain the following corollary.

Corollary 31 *For the metric based on any ℓ_p -norm with $1 \leq p < \infty$, there exists an input set $X \subset \mathbb{R}^d$ with $d = k + \log_2 k$ such that Algorithm 3 computes a solution to Problem 3 with $\sqrt[p]{\frac{1}{2} \log_2 k}$ times the cost of an optimal solution.*

Additionally, considering the diameter k -clustering problem with respect to the ℓ_∞ -metric, it is known that every n -point subset of an arbitrary metric space can be embedded isometrically into $(\mathbb{R}^n, \ell_\infty)$ [8]. Hence, the instance from Proposition 30 of size $n = k^2$ yields an instance in \mathbb{R}^{k^2} satisfying the same approximation bound with respect to the ℓ_∞ -distance. We obtain the following corollary.

Corollary 32 *For the metric based on the ℓ_∞ -norm, there exists an input set $X \subset \mathbb{R}^d$ with $d = k^2$ such that Algorithm 3 computes a solution to Problem 3 with $\frac{1}{2} \log_2 k$ times the cost of an optimal solution.*

4.4.1 The Discrete k -Center Problem

The input instance X from Proposition 30 also proves lower bounds on the approximation factor of the agglomerative solution to the discrete k -center problem. To this end, just note that for the instance X in every step of the algorithm the minimal discrete radius of a cluster equals the diameter of the cluster. We immediately obtain the following corollaries.

Corollary 33 *For the metric based on any ℓ_p -norm with $1 \leq p < \infty$, there exists an input set $X \subset \mathbb{R}^d$ with $d = k + \log_2 k$ such that Algorithm 1 computes a solution to Problem 1 with $\sqrt[p]{\frac{1}{2} \log_2 k}$ times the cost of an optimal solution.*

Corollary 34 *For the metric based on the ℓ_∞ -norm, there exists an input set $X \subset \mathbb{R}^d$ with $d = k^2$ such that Algorithm 1 computes a solution to Problem 1 with $\frac{1}{2} \log_2 k$ times the cost of an optimal solution.*

Moreover, in case of the ℓ_2 -norm, we obtain the following corollary.

Corollary 35 *For the metric based on the ℓ_2 -norm, there exists an input set $X \subset \mathbb{R}^d$ with $d = O(\log^3 k)$ such that Algorithm 1 computes a solution to Problem 1 with $\Omega(\sqrt{\log k})$ times the cost of an optimal solution.*

Corollary 35 follows by embedding the instance from Corollary 33 into the $O(\log^3 k)$ -dimensional Euclidean space without altering the behavior of the agglomerative algorithm or the lower bound of $\Omega(\sqrt{\log k})$ (Johnson-Lindenstrauss embedding [10]). For this embedded instance, the bound given in Sect. 3.1 states an upper bound of $20d + 2 \log(k) + 2 = O(\log^3 k)$ times the cost of an optimal solution. Hence, in case of the discrete k -center clustering using the ℓ_2 -metric, the upper bound from our analysis almost matches the lower bound.

Furthermore, this implies that the approximation factor of Algorithm 1 cannot be simultaneously independent of d and $\log k$. More precisely, the approximation factor cannot be sublinear in $\sqrt[6]{d}$ and in $\sqrt{\log k}$.

5 Open Problems

The main open problems our work raises are:

- Can the doubly exponential dependence on d in Theorem 21 be improved?
- Are the different dependencies on d in the approximation factors for the discrete k -center problem, the k -center problem, and the diameter k -clustering problem due to the limitations of our analysis or are they inherent to these problems?
- Can our results be extended to more general distance measures?
- Can the lower bounds for ℓ_p -metrics with $1 < p < \infty$ be improved to $\Omega(\log k)$, matching the lower bound from [4] for all ℓ_p -norms?

References

1. Bădoiu, M., Har-Peled, S., Indyk, P.: Approximate clustering via core-sets. In: Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing, STOC'02, pp. 250–257. ACM, New York (2002)
2. Broder, A.Z., Glassman, S.C., Manasse, M.S., Zweig, G.: Syntactic clustering of the web. *Comput. Netw. ISDN Syst.* **29**(8–13), 1157–1166 (1997). Papers from the Sixth International World Wide Web Conference
3. Charikar, M., Chekuri, C., Feder, T., Motwani, R.: Incremental clustering and dynamic information retrieval. In: Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing, STOC'97, pp. 626–635. ACM, New York (1997)
4. Dasgupta, S., Long, P.M.: Performance guarantees for hierarchical clustering. *J. Comput. Syst. Sci.* **70**(4), 555–569 (2005). Special Issue on COLT 2002
5. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci.* **95**(25), 14863–14868 (1998)
6. Feder, T., Greene, D.: Optimal algorithms for approximate clustering. In: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, STOC'88, pp. 434–444. ACM, New York (1988)
7. Florek, K., Lukaszewicz, J., Perkal, J., Steinhaus, H., Zubrzycki, S.: Sur la liaison et la division des points d'un ensemble fini. *Colloq. Math.* **2**, 282–285 (1951)
8. Fréchet, M.: Les dimensions d'un ensemble abstrait. *Math. Ann.* **68**(2), 145–168 (1910)
9. Gonzalez, T.F.: Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.* **38**(0), 293–306 (1985)
10. Johnson, W.B., Joram, L.: Extensions of Lipschitz mappings into a Hilbert space. In: Conference in Modern Analysis and Probability. Contemporary Mathematics, vol. 26, pp. 189–206. Am. Math. Soc., Providence (1984)
11. Lee, K., Kim, J., Hoon Kwon, K., Han, Y., Kim, S.: DDoS attack detection method using cluster analysis. *Expert Syst. Appl.* **34**(3), 1659–1665 (2008)
12. McQuitty, L.L.: Elementary linkage analysis for isolating orthogonal and oblique types and typical relevancies. *Educ. Psychol. Meas.* **17**, 207–209 (1957)
13. Naszódi, M.: Covering a set with homothets of a convex body. *Positivity* **14**, 69–74 (2010)
14. Pereira, F., Tishby, N., Lee, L.: Distributional clustering of English words. In: Proceedings of the 31st Annual Meeting on Association for Computational Linguistics, ACL'93, pp. 183–190. Association for Computational Linguistics, Stroudsburg (1993)
15. Sneath, P.H.A., Sokal, R.R.: Numerical Taxonomy: the Principles and Practice of Numerical Classification. Freeman, New York (1973)
16. Webster, R.: Convexity. Oxford Science Publications. Oxford University Press, London (1994)