Approximations for Hierarchical and Lower-Bounded Clustering and the Complexity of Minimum-Error Triangulation

DISSERTATION

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Anna Arutyunova

aus

Visaginas, Litauen

Bonn 2023

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

Gutachter/Betreuer:	Prof. Dr. Heiko Röglin
Gutachterin:	Prof. Dr. Melanie Schmidt

Tag der Promotion: 14.12.2023
Erscheinungsjahr: 2024

Abstract

Clustering deals with the problem of finding structures in data. Given a number k we search for a good partition of a set of data points into at most k sets, where the sets of the partition are called clusters. In the hierarchical clustering problem we have to find a sequence of clusterings, one for every possible number k, such that the clusterings in the sequence are hierarchically compatible with each other. We say that such a hierarchical clustering achieves an α-approximation with respect to a given cost function, such as the radius (k-center), the sum of (squared) distances between points and their closest center (k-median and k-means), if the following holds. For every k the cost of its k-clustering can be bounded by α times the cost of an optimal clustering of size k. However for most of the popular k-clustering objectives there does not exist a hierarchical clustering with approximation factor $\alpha = 1$ since the optimal clusterings are in general not hierarchically compatible. Therefore we are interested in the smallest α such that there always exists a hierarchical clustering with approximation factor α. We construct a clustering instance which shows that for the objective discrete radius (drad) we have $\alpha \geq 4$ and for radius (rad) and diameter (diam) we have $\alpha \geq 3 + 2\sqrt{2}$. For drad we already know that we can always compute a hierarchical clustering with approximation factor 4 and we show that there always exists a hierarchical clustering with approximation factor $3 + 2\sqrt{2}$ for rad and diam.

In practice agglomerative clustering methods are often used to compute hierarchical clusterings. One such algorithm is the complete linkage algorithm, which starts with the clustering where every point is in its own cluster and merges the two clusters whose union has the smallest (discrete) radius or diameter among all possible merges until all points are in the same cluster. We know that this algorithm does compute an $O(1)$-approximation in the case where the metric space is Euclidean and of constant dimension. For general metric spaces we only know that the approximation factor of the algorithm is in $\Omega(\log(k))$. We show that the approximation factor is in fact in $\Omega(k)$ for discrete radius and diameter and that complete linkage computes an $O(k)$-approximation for discrete radius and an $O(k^{1.59})$-approximation for diameter.

We further consider k-median/k-means with lower bounds, where we need to partition the set of data points into at most k clusters of size at least B for a given bound $B \in \mathbb{N}$ while minimizing the k-median or k-means objective. There already exist $O(1)$-approximation algorithms for the related facility location problem with lower bounds, which can be transformed into an $O(1)$-approximation algorithm for k-median with lower bounds. We present a general approach that shows how to transform any approximation algorithm for facility location with lower bounds into an approximation algorithm for k-median with lower bounds. Since the resulting approximation factors are high, we consider a relaxed variant of lower bounds, which we call weak lower bounds. Here we allow that points can be contained in multiple clusters. We discuss how to compute a $(6.5 + \epsilon)$-approximation for k-median with weak lower bounds and an $O(1)$-approximation for k-means with weak lower bounds. Furthermore we show that we can transform every solution with weak lower bounds into a solution with 2-weak lower bounds, where every point is contained in at most two clusters, while the cost only increases by a constant factor. We generalize this result and show that in fact we can achieve $(1 + \epsilon)$-weak lower bounds, where every point is assigned to one center and possibly a fraction of ϵ to another center. Finally we present a bi-criteria approximation algorithm for k-means with lower bounds.

In the last part of this thesis we study the minimum-error triangulation problem. Given a set of triangulation points $\mathcal{S} \subset \mathbb{R}^2$ and values $f\colon \mathcal{S} \to \mathbb{R}$ and a set of reference points $\mathcal{R}$ in the convex hull of $\mathcal{S}$ with values $h\colon \mathcal{R} \to \mathbb{R}$, we want to find a triangulation of $\mathcal{S}$ such that the linear interpolation of f on the triangulation represents the values at $\mathcal{R}$ as well as possible, i.e., the sum of squared differences between the values at $\mathcal{R}$ and their interpolation is minimized. This problem has been studied before in the context of sea surface reconstruction. While there exist many heuristics to compute good minimum-error triangulations it was not known whether there exist polynomial time approximation algorithms for this problem. We show that the problem to decide whether there exists a triangulation of zero error is already NP-hard, which implies that there does not exist an approximation algorithm for minimum-error triangulation which runs in polynomial time unless $\mathsf{P} = \mathsf{NP}$.

Acknowledgments

In the following I would like to thank all people who made this thesis possible.

First of all I would like to thank my supervisor Heiko Röglin who supported me and provided guidance during the course of my PhD. I was welcomed in a pleasant work environment and was able to work freely on research topics that I was interested in. I would also like to thank Melanie Schmidt who supported me and always comes up with new interesting problems.

I am grateful for all the nice colleagues from our department that I had the pleasure to meet. I always felt welcome and I am really happy that I could get to know you all better after the long lasting home-office period. From my first months I especially remember the 4 pm math-calendar problems solving meetings before Christmas. Later on the discussions about games, music, food and many other topics as well as the cooking and board game events made my time with you very enjoyable.

I would also like to thank all of my co-authors, Anne Driemel, Anna Großwendt, Jan-Henrik Haunert, Herman Haverkort, Jürgen Kusche, Elmar Langetepe, Philip Mayer, Petra Mutzel, Heiko Röglin, Melanie Schmidt, and Julian Wargalla.

During my time as PhD student I was financially supported by the German Research Foundation (DFG) - project number 416767905.

Finally I would like to thank my friends and family: my mother and sister, as well as Amanda, who always supported me in many ways and provided the distraction from work that I sometimes needed. Especially I thank Michael. You not only read parts of this thesis and listened to me whenever I talked about my research but you were always there for me with your incredible kindness and care.

Contents

Chapter 1

Introduction

Clustering is an unsupervised learning tool for finding structures in data and has been extensively used and studied in the last decades. Usually we are given a set of data points and a measure which tells us how similar or dissimilar two data points are. The task is then to compute a partition of the data into a certain number of subsets, or clusters as we call them from now on. Naturally we would like that similar objects are contained in the same cluster while dissimilar objects are contained in different clusters.

Clustering can be modeled in many different ways. We focus on a popular theoretical model where the data is represented by points in a metric space, for example the Euclidean space, and that the number k of clusters is part of the input. The metric then indicates how similar or dissimilar two points are. If two points have a small distance to each other, they are assumed to be similar and are more likely to be contained in the same cluster. Often we additionally have to choose a center out of the metric space for every cluster that represents the points in the cluster well. Then the goal is to minimize some cost function like the maximum radius or diameter of a cluster (k-center/k-diameter) or the sum of (squared) distances of points to their centers (k-median/k-means). We call a solution which has the smallest cost among all possible clusterings optimal.

As an example we can think of the problem of placing k shops in some given area. In this scenario the points represent households and the task is to find good locations to build k shops which serve the households well. If we want that there is no person which has to travel too far to its nearest shop, we would like to minimize the maximum traveling distance between a household and its nearest shop. This can be modeled by the k-center problem. Another possibility is to minimize the total traveling distance of all persons. So there may be persons which have a large traveling distance to a shop but we guarantee that a large fraction of persons do not travel a long distance to their nearest shop. This can be modeled by sum based cost functions such as k-median and k-means.

In general we cannot expect to compute an optimal solution efficiently for any of these problems unless $\mathsf{P} = \mathsf{NP}$. Here efficient means that the running time of the algorithm is polynomial in the input size of the problem instance. Therefore we focus on computing a solution which is close to the optimal solution. If OPT denotes the cost of an optimal clustering and $\alpha > 1$, we want to compute a solution of cost at most $\alpha\mathsf{OPT}$ efficiently. Such solutions are called α-approximations. For k-center there exists an efficient algorithm which computes such a solution for any instance for $\alpha = 2$ [43]. For k-median and k-means there exist such algorithms for $\alpha = 2.67059 + \epsilon$ [32] and $\alpha = 5.912 + \epsilon$ [31], respectively.

For some applications it is desirable to only allow clusters which have some additional properties. For example given some number B we can ask for a clustering where every cluster consists of at least B points. This problem is known as clustering with lower bounds. Similarly we can enforce the clusters to have a size of at most B, which is known as clustering with upper bounds. In the above example where we have to place shops this could be useful to model that it is only economically viable to open a shop if it serves at least B households (lower bounds) or to model that it has a certain capacity of goods and therefore can only serve at most B households (upper bounds).

It is usually harder to compute reasonably good solutions for clustering under side constraints efficiently. In the unconstrained version we make the problem easier by allowing for solutions which are not necessarily optimal with respect to the given cost function but have a cost that is within a factor α of the optimal cost. Similarly we can relax the side constraints. For clustering with lower bounds we allow that clusters violate the lower bound condition by some factor $\beta \in (0, 1)$, i.e., every cluster has to contain βB instead of B points. In a similar way we can relax upper bound constraints. This leads to the concept of bi-criteria approximations. For clustering with lower bounds we call a solution an (α, β)-approximation if every cluster contains at least βB points and has cost at most $\alpha\mathsf{OPT}$. Here OPT denotes the cost of the best clustering where every cluster consists of at least B points. Another possibility to relax the lower bound constraint is to allow that a point can be contained in multiple clusters. While this may increase the cost induced by a point, this allows us to satisfy the lower bound constraints more easily. We call this clustering with weak lower bounds and will see how one can obtain reasonably good solutions for this relaxed version of the problem.

However, all of these problems require that we know the number of clusters k beforehand. It may be already a non-trivial task to determine the right number of clusters for some given data set and there may be multiple reasonable values for k. Suppose that the data consists of DNA sequences of different animals. It is not hard to picture that there are multiple values of k which produce sensible clusterings, for example for different taxonomic ranks as order, family or species. We expect that the data admits a hierarchical structure which allows to group the animals on different levels of granularity (e.g. the taxonomic ranks). This leads us to the concept of hierarchical clustering. A hierarchical clustering of a data set is actually a sequence of clusterings, one for each possible number of clusters. We require that every two clusterings are hierarchically compatible: Given a k-clustering and a k'-clustering with $k < k'$, the two clusterings are hierarchically compatible if for every cluster in the k'-clustering there is a cluster in the k-clustering containing this cluster.

A hierarchical clustering provides information on different levels of granularity. However, it is not clear how to evaluate the quality of such clusterings. A way to evaluate the quality is to compare the cost of the k-clustering with the cost of an optimal k-clustering for every k. This has the advantage that we can guarantee a good cost on every level. Roughly speaking given some $\alpha \geq 1$ we want to find a hierarchical clustering which is an α-approximation, i.e., the cost of its k-clustering is worse by factor at most α than the cost of an optimal k-clustering for every k. However, it may happen that the optimal clusterings are not hierarchically compatible, so we can generally not expect that there exists a hierarchical clustering which is optimal for every possible number of clusters.

There are two common ways to compute hierarchical clusterings. The first possibility is to start with the clustering in which every data point forms its own cluster and then successively merge two clusters until all points are in the same cluster. It is easy to see

that the result is a hierarchical clustering. Usually the decision which two clusters we merge in each step depends on the cost function which is to be optimized. These kinds of approaches are known as agglomerative clustering methods. The second possibility is to construct clusterings the other way around. We start with the clustering of size one which consists of a cluster containing all points and successively divide a cluster into two clusters until every point forms its own cluster. These kinds of approaches are known as divisive clustering methods.

We study the performance of the popular complete linkage algorithm, which is an agglomerative clustering method. In every step complete linkage merges the two clusters whose merge results in the smallest radius (or diameter). We are interested in bounding the ratio between the radius (or diameter) of the k-clustering computed by complete linkage and the radius (or diameter) of the optimal k-clustering solution for every possible k.

Besides the analysis of the approximation guarantee of popular algorithms for hierarchical clustering, we are interested in the smallest α such that there exists a hierarchical clustering with approximation factor α for every clustering instance. We call this α the *price of hierarchy*.

In the last part of this thesis we focus on the minimum-error triangulation problem. This problem is considered in the work by Nitzke et al. [68] who focus on computing triangulations of the sea surface by taking into account two types of data. The first type of data is altimeter data obtained from tide gauge stations located at the sea shore. The second type of data is altimeter data of the sea surface measured by satellites. The tide gauge stations serve as triangulation points which are used to triangulate the sea surface. Every triangulation of the sea surface now induces a piece-wise linear map by linearly interpolating the altimeter data at the tide gauge stations in every triangle. A triangulation is considered good if the interpolation of the tide gauge altimeter data represents the satellite altimeter data on the sea surface well. Data from the tide gauge stations is available for a long period of time, starting in the late 17th century for some of the tide gauge stations while satellite altimeter data is only available since 1993. Suppose we learn a good triangulation of the sea surface at some given year, then it is a reasonable assumption that the learned triangulation of the sea surface together with the altimeter data from tide gauge stations will recover the altimeter data on the sea surface for years where satellite altimeter data is not available. Using integer linear programming techniques Nietzke et al. [68] learn such a triangulation of the sea surface using altimeter data from tide gauge stations and satellite altimeter data at some fixed time and evaluate their triangulation at other times where both altimeter data from tide gauge stations and satellite altimeter data are available. However, the question remains whether we can expect to find efficient (approximation) algorithms which compute such minimum-error triangulations. We will see later that unless $\mathsf{P} = \mathsf{NP}$ we cannot expect to find such algorithms.

1.1 Clustering Objectives

The k-clustering problem can be defined as follows: We are given a metric space $(\mathcal{X}, d)$, a set of n points $\mathcal{P} \subset \mathcal{X}$ and an objective function which assigns to every k-clustering a positive real number. A k-clustering is a partition of the set $\mathcal{P}$ into k subsets $\mathscr{C} = (\mathcal{C}_1, \ldots, \mathcal{C}_k)$. For an objective function cost we denote by $\mathcal{O}_{\mathsf{cost}}$ the optimal solution with

respect to the objective cost. The task is to compute a k-clustering which approximates the optimal solution as well as possible.

We consider the following objective functions throughout this thesis.

(discrete) k-center: The *discrete radius* of a cluster $\mathcal{C} \subset \mathcal{P}$ is defined as $\mathsf{drad}(\mathcal{C}) = \min_{c \in \mathcal{C}} \max_{x \in \mathcal{C}} d(c, x)$. Thus it is the radius of the cluster with respect to the best possible center chosen from $\mathcal{C}$. The cost of a k-clustering $\mathscr{C}$ is then given as $\mathsf{drad}(\mathscr{C}) = \max_{\mathcal{C} \in \mathscr{C}} \mathsf{drad}(\mathcal{C})$, the maximum radius of a cluster in $\mathscr{C}$.

(non-discrete) k-center: The *radius* of a cluster $\mathcal{C} \subset \mathcal{P}$ is defined as $\mathsf{rad}(\mathcal{C}) = \min_{c \in \mathcal{X}} \max_{x \in \mathcal{C}} d(c, x)$. Thus it is the radius of the cluster with respect to the best possible center chosen from $\mathcal{X}$. The cost of a k-clustering $\mathscr{C}$ is then given as $\mathsf{rad}(\mathscr{C}) = \max_{\mathcal{C} \in \mathscr{C}} \mathsf{rad}(\mathcal{C})$.

k-diameter: The *diameter* of a cluster $\mathcal{C} \subset \mathcal{P}$ is defined as $\mathsf{diam}(\mathcal{C}) = \max_{x,y \in \mathcal{C}} d(x, y)$. Thus it is the maximum distance between two points in the same cluster. The cost of a k-clustering $\mathscr{C}$ is then given as $\mathsf{diam}(\mathscr{C}) = \max_{\mathcal{C} \in \mathscr{C}} \mathsf{diam}(\mathcal{C})$.

The only difference between radius and discrete radius is the restriction that for the discrete radius the center has to be contained in the cluster itself. While this may be a small detail this has actually an impact on the price of hierarchy as well as on the approximation guarantee of complete linkage as we see later.

All of these problems are related to each other. For the optimal solutions it holds that $\mathsf{rad}(\mathcal{O}_{\mathsf{rad}}) \leq \mathsf{diam}(\mathcal{O}_{\mathsf{diam}}) \leq 2\mathsf{rad}(\mathcal{O}_{\mathsf{rad}})$ and for any k-clustering $\mathscr{C}$ we obtain a similar inequality $\mathsf{rad}(\mathscr{C}) \leq \mathsf{diam}(\mathscr{C}) \leq 2\mathsf{rad}(\mathscr{C})$. Therefore a k-clustering $\mathscr{C}$ which is an α-approximation with respect to rad is a 2α-approximation for the diam objective and vice versa. This is also true for the combinations diam, drad and rad, drad.

One of the easiest approximation algorithms for a clustering problem is probably the algorithm of Gonzales [43] for k-center and k-diameter. It first computes an enumeration of points in $\mathcal{P}$ as follows. The first point p_1 is chosen arbitrarily from the set $\mathcal{P}$, the second point p_2 is then the point with largest distance to p_1. If we determined $p_1, \ldots, p_i$ already then the $(i+1)$-st point is the point which maximizes $\min_{1 \leq j \leq i} d(p, p_j)$ for $p \in \mathcal{P}$. Thus p_{i+1} is the point in $\mathcal{P}$ farthest away from the first i points.

Now we can use the enumeration of points in $\mathcal{P}$ to determine a k-clustering. We assign every point in $\mathcal{P}$ to the nearest point from $p_1, \ldots, p_k$. Then two points are in the same cluster if they are assigned to the same point. This yields a k-clustering that is a 2-approximation with respect to the cost functions $\mathsf{rad}, \mathsf{drad}, \mathsf{diam}$. Since it is NP-hard to compute α-approximations for all of these objectives for $\alpha < 2$ [52] this algorithm already achieves the best possible approximation guarantee. Another well-known 2-approximation algorithm for these objectives is the algorithm by Hochbaum and Shmoys [52].

When talking about sum-based objectives we usually define a k-clustering not as a partition of the set $\mathcal{P}$ but as a set of k centers $C \subset \mathcal{X}$ and an assignment $\sigma \colon \mathcal{P} \to C$ of points to centers. Let $C = \{c_1, \ldots, c_k\}$, then this gives us a partition of the set $\mathcal{P}$ into k subsets $\mathscr{C} = (\mathcal{C}_1, \ldots, \mathcal{C}_k)$ with $\mathcal{C}_i = \sigma^{-1}(c_i)$. Thus both definitions are related to each other. The most popular sum-based objectives are k-median and k-means.

k-median: This objective is a sum over the distances between points and their assigned centers $\mathsf{med}(C, \sigma) = \sum_{x \in \mathcal{P}} d(x, \sigma(x))$.

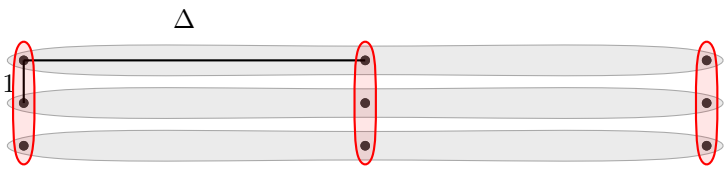

Figure 1.1: An instance for 3-means consisting of 9 points. A possible solution of Lloyd's algorithm is depicted in gray and the optimal solution is depicted in red.

k-means: Similarly to the k-median objective we take the sum over the *squared* distances between points and their assigned centers $\mathsf{mean}(C, \sigma) = \sum_{x \in \mathcal{P}} d^2(x, \sigma(x))$.

Usually these problems, especially k-means, are studied in the Euclidean space. Here we consider the more general version where $(\mathcal{X}, d)$ can be any metric space. Approximation algorithms for k-median and k-means have been extensively studied. In the case of an arbitrary metric space we know that both problems are NP-hard to approximate within a factor of $1 + 2/e$ for k-median and $1 + 8/e$ for k-means [48]. The best known approximation algorithms achieve an approximation factor of $2.67059 + \epsilon$ [32] for k-median and $9 + \epsilon$ [54] for k-means. If the underlying metric space is Euclidean then we know that both problems are NP-hard to approximate within a factor of 1.015 for k-median and 1.06 for k-means [34] while the best known approximation algorithms achieve an approximation guarantee of $2.406 + \epsilon$ for k-median and $5.912 + \epsilon$ for k-means [31]. However, these algorithms are very complex. Therefore, in practice simpler algorithms are used, even though they do not have good theoretical performance. For instances in the Euclidean space Lloyd's algorithm [64] has become a popular heuristic to compute solutions for k-means. It starts with k centers chosen arbitrarily from $\mathcal{P}$ and is based on two optimization steps. The first step is the following: If we are given a set of k centers, we compute the respective k-clustering by assigning every point to its closest center. In the second step we then optimize over the choice of centers, i.e., for a cluster we define the new center to be the point which minimizes the sum of squared distances to points in the cluster. It is a well-known fact that the best choice for the new center is the mean of points in the cluster. Then we proceed with the first and second optimization step until the solution stabilizes. While this works well in practice, the running time of Lloyd's algorithm can be exponential in the worst case [8] and the computed solution can be arbitrarily bad. To see that the solution can be arbitrary bad we consider a set of 9 points on the plane with coordinates $(0,0), (0,1), (0,-1), (-\Delta,0), (-\Delta,1), (-\Delta,-1), (\Delta,0), (\Delta,1), (\Delta,-1)$ as illustrated in Figure 1.1. If we want to compute a 3-means solution with Lloyd's algorithm we may choose $(0,0), (0,1), (0,-1)$ as centers in the beginning. Now assigning every point to its closest center and recomputing the means of the clusters will result again in the centers $(0,0), (0,1), (0,-1)$. The respective solution has then cost $6\Delta^2$ while the optimal solution chooses $(-\Delta,0), (0,0), (\Delta,0)$ as centers and has cost 6. Since we can choose Δ arbitrarily large, this shows that the approximation factor of Lloyd's algorithm can be arbitrarily bad.

The k-means++ algorithm improves upon the choice of the k centers in the beginning of Lloyd's algorithm. It samples the first center uniformly at random from $\mathcal{P}$, then the i-th center is sampled proportionally to the squared distance to the first $i-1$ centers, so if a point is very far away from the centers sampled so far, the probability to sample this center is high. After sampling the k centers we proceed with Lloyd's algorithm. The

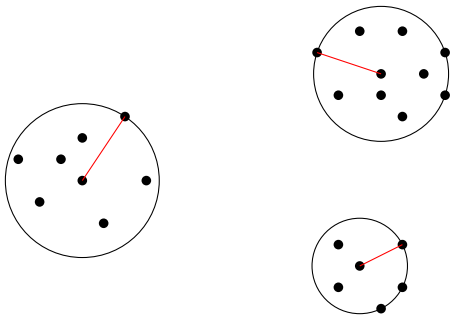

Figure 1.2: Here we see a possible 4-center solution with lower bound $B = 6$. As we see it is better to take a clustering of size three since any solution with four clusters results in clusters of larger radius to satisfy the lower bound.

k-means++ algorithm computes an $O(\log(k))$-approximation in expectation [9].

Facility location is not exactly a clustering problem but closely related to k-median clustering. We are given a set of clients $\mathcal{D}$, a set of facilities $\mathcal{F}$ with facility opening costs $f_i \geq 0$ for every $i \in \mathcal{F}$ and a metric d on $\mathcal{D} \cup \mathcal{F}$. The task is to compute a set $C \subset \mathcal{F}$ of open facilities and an assignment $\sigma \colon \mathcal{D} \to C$ such that the following objective is minimized:

facility location: The objective is a sum of the opening cost of facilities and the sum of distances between clients and their assigned facility $\mathsf{fac}(C, \sigma) = \sum_{i \in C} f_i + \sum_{p \in \mathcal{D}} d(p, \sigma(p))$.

The facility location problem is very similar to the k-median problem with the only difference that the number of centers is not bounded by k but instead we pay a certain cost for every center we choose. Li [60] presents the currently best approximation algorithm with an approximation ratio of 1.488, while the best known lower bound is 1.463 [48] unless $\mathsf{P} = \mathsf{NP}$.

1.2 Clustering with Weak Lower Bounds

In the first part of this thesis we study k-clustering problems with lower bound constraints. Suppose we are given a metric space $(\mathcal{X}, d)$, a set of n points $\mathcal{P} \subset \mathcal{X}$ and a number $1 \leq B \leq n$. We are interested in computing a clustering consisting of at most k clusters such that the size of every cluster is at least B, while minimizing some objective function like k-center, k-median or k-means. Notice that here we explicitly allow clusterings that consist of fewer than k clusters. We can define lower-bounded facility location in a similar way by enforcing that every open facility serves at least B clients.

Enforcing a lower bound on the size of a cluster can be of interest for example if the data points contain sensitive information, so we want to publish only a representative (e.g. the center) and the size of each cluster. If now the clusters are sufficiently large this guarantees a certain level of anonymity for the individual data points. This idea was proposed by Aggarwal et al. [3] as a motivation to study k-center with lower bounds.

Lower bound constraints have been studied before in the context of radii-based clustering and facility location. If we consider the k-center objective, we first notice that in the unconstrained version of the problem the cost of an optimal solution decreases

with growing k, this may not be the case for clustering with lower bounds as we see for example in Figure 1.2. If k is too large it may be even impossible to find solutions with k clusters which satisfy the lower bound constraint at every cluster. Therefore we also may remove the constraint of having at most k clusters and instead ask for a clustering which minimizes the maximum radius and may have arbitrarily many clusters of size at least B. Aggarwal et al. [3] studied both versions of the problem, one where the number of clusters is upper bounded by some number k, i.e., k-center with lower bounds, and one where the number of clusters can be arbitrarily large. For both versions they give a polynomial time 2-approximation. In the more general version of this problem the lower bound is not uniform but every point $p \in \mathcal{X}$ has its own lower bound $B(p)$, which has to be satisfied if p is chosen as center of a cluster. For this version Ahmadian and Swamy [5] present a 3-approximation.

Regarding sum based objectives Karger and Minkoff [55] as well as Guha, Meyerson and Munagala [49] show how to compute bi-criteria solutions for lower-bounded facility location, i.e. solutions that can violate the lower bound by a certain factor and whose cost can be bounded in terms of the cost of an optimal solution. The first algorithm which computes a constant factor approximation to lower-bounded facility location is the algorithm by Svitkina [72]. It first computes a bi-criteria solution as in [55, 49]. To decide which facilities to remove from the solution to satisfy the lower bound constraints, Svitkina reduces the instance to an instance of capacitated facility location, for which constant factor approximations are already known [19]. The result is a 558-approximation, which was later improved to 82.6 by Ahmadian and Swamy [4]. As in radii-based clustering we can consider a more general version of lower-bounded facility location which allows for an individual lower bound $B(i)$ for every facility i. Li [61] proves that there is a 4000-approximation for this problem, which we call non-uniform lower-bounded facility location. There also has been an effort to design algorithms with better approximation guarantees which run in FPT time. Bera et al. [20] give a 4.676-approximation for k-median with lower bounds running in time $2^k\text{poly}(n)$. For Euclidean instances in $\mathbb{R}^d$, Bandyapadhyay et al. [18] give a $(1+\epsilon)$-approximation for k-median and k-means with lower bounds running in time $2^{\widetilde{O}(k/\epsilon^{O(1)})}\text{poly}(nd)$.

We are interested in k-median/k-means clustering with lower bounds. Most of the algorithms known for facility location with lower bounds can be transformed into algorithms for k-median with lower bounds while the approximation factor can increase by a constant factor. The bi-criteria algorithm for lower-bounded facility location in [55, 49] can be transformed relatively straightforwardly to a bi-criteria algorithm for k-median with lower bounds [50]. Furthermore the algorithms [72, 4] for lower-bounded facility location can be adapted for k-median by replacing the first reduction step. This is discussed in more detail in [51]. We show that using the concept of nesting, which is for example also used in the design of approximation algorithms for hierarchical clustering [63], any approximation algorithm for facility location with lower bounds can be transformed into an approximation algorithm for k-median with lower bounds and this reduction works also for more general k-clustering problems including k-means.

Since the approximation factor of algorithms for facility location with lower bounds is relatively high this yields high approximation factors for k-median with lower bounds as well. Also it is not clear whether the results in [72, 4, 49, 55] can be adapted for k-means since the triangle inequality does not hold for squared Euclidean metrics, which causes a problem when bounding the cost of the computed clustering.

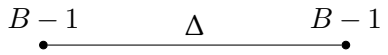

Figure 1.3: On the difference between lower-bounded clustering and weakly lower-bounded clustering.

Therefore we consider a relaxation of this problem which we call *weakly lower-bounded* k-clustering where every point may be part of multiple clusters. Now our solution does not consist of a partitioning of $\mathcal{P}$, instead we want to cover $\mathcal{P}$ by at most k clusters, each of size at least B, and these clusters may intersect. To explain this idea, consider Figure 1.3. The instance consists of two sets of $(B-1)$ points such that the distance of two points in the same set is small and the distance of two points in different sets is large. For simplicity we assume that the distance between points in the same set is zero. In Figure 1.3 we have two locations with $B-1$ points, and the distance between the two locations is Δ. If we enforce a lower bound of B, we can open only one center, which results in a cost of $(B-1)\Delta$ for 2-median and $\Omega(B\Delta^2)$ for 2-means. If we allow points to be assigned to multiple centers we can open two centers, one at each of the two locations, and assign one point from every location to both centers and all other points to its nearest center. This yields clusters of size B and the cost reduces from $(B-1)\Delta$ to 2Δ for 2-median and from $\Omega(B\Delta^2)$ to $\Omega(\Delta^2)$ for 2-means. Thus even if we have to pay additional connection cost for two of the points, the overall cost is smaller. In conclusion we see that the cost of an optimal solution for clustering with lower bounds cannot be bounded by the cost of an optimal solution for clustering with weak lower bounds within a constant factor.

1.2.1 Results

We obtain the following results. First we show that any solution which satisfies lower bounds but has arbitrary many clusters can be transformed into a solution with at most k clusters.

Corollary (2.2.4)**.** *Suppose there exists a λ-approximation algorithm for k-median and a μ-approximation algorithm for facility location with uniform or non-uniform lower bounds. Then there exists a $(2\mu+\lambda)$-approximation for k-median with uniform lower bounds and a $(3\mu+2\lambda)$-approximation for k-median with non-uniform lower bounds.*

Then we focus on weakly lower-bounded k-median/k-means. Both problems can be reduced relatively straightforwardly to k-median/k-means with center costs, i.e., a variant where opening a center c induces a cost f_c. Since there exists a $(3.25+\epsilon)$-approximation for k-median with center costs [28] and an $O(1)$-approximation for k-means with center costs [53], we obtain a $(6.5+\epsilon)$-approximation for weakly lower bounded k-median and an $O(1)$-approximation for weakly lower-bounded k-means.

We show that we can transform every solution for weakly lower-bounded k-median/k-means into a solution where every point is assigned to at most two centers, which we call solutions with 2-weak lower bounds. Both k-median and k-means are special cases of the generalized k-median problem where the distance function d satisfies all properties of a metric but the triangle inequality is only satisfied with a factor α, i.e., for all $x, y, z \in \mathcal{X}$ we have $d(x,z) \leq \alpha(d(x,y)+d(y,z))$. The cost of a solution to this problem equals the k-median cost med since it is just the sum of distances between points an their assigned centers.

Theorem (2.3.4)**.** *Given a solution (C, σ) to generalized k-median with weak lower bounds, we can compute a solution $(\widetilde{C}, \widetilde{\sigma})$ to generalized k-median with 2-weak lower bounds (assigning every point at most twice) in polynomial time such that* $\mathsf{med}(\widetilde{C}, \widetilde{\sigma}) \leq \alpha(\alpha+1)\,\mathsf{med}(C, \sigma)$.

If we plug in $\alpha = 1$ for k-median, we obtain a solution for k-median with 2-weak lower bounds which is by factor 2 worse than the given solution for k-median with weak lower bounds. For k-means we plug in $\alpha = 2$ and get an increase by a factor of 6.

Remember that lower-bounded clustering can be of interest when the points contain sensitive information which we want to protect. The lower bound guarantees that publishing the centers of the clusters does not reveal information about the individual data points. We also can use 2-weak lower-bounded clustering to guarantee anonymity since the lower bound is satisfied for every cluster and at the same time the data distortion is not to large, since every point is allowed to be contained in at most two clusters. We can reduce the distortion of the data set even further by allowing a point to be assigned to one center and with a fraction $\epsilon \in (0, 1)$ to a second center. We say that such solutions satisfy $(1+\epsilon)$-weak lower bounds.

Theorem (2.3.10)**.** *Given $0 < \epsilon < 1$ and a solution (C, σ) to generalized k-median with weak lower bounds, we can compute a solution $(\widetilde{C}, \widetilde{\sigma})$ to generalized k-median with $(1+\epsilon)$-weak lower bounds in polynomial time such that* $\mathsf{med}(\widetilde{C}, \widetilde{\sigma}) \leq (\lceil \frac{1}{\epsilon} \rceil \alpha(\alpha+1) + 1)\mathsf{med}(C, \sigma)$.

Finally, we show that our results imply an $(O(1), O(1))$-bi-criteria approximation algorithm for generalized k-median with lower bounds, where the lower bounds are satisfied only to an extent of $B/O(1)$.

Theorem (2.4.2)**.** *Given a γ-approximate solution (C, σ) to generalized k-median with 2-weak lower bounds and a fixed $\beta \in [0.5, 1)$ we can compute a $(\beta, \gamma \max\{\frac{\alpha\beta}{1-\beta} + 1, \frac{\alpha^2\beta}{1-\beta}\})$-bi-criteria solution to generalized k-median with lower bounds in polynomial time. In particular, there exists a polynomial time $(\frac{1}{2}, O(1))$-bi-criteria approximation algorithm for k-means with lower bounds.*

Notice that there already exists a bi-criteria algorithm for facility location [49, 55] which can be transformed easily into a bi-criteria algorithm for k-median. However, the triangle inequality is crucial for the analysis of this algorithm, therefore it is not clear if it can be used to compute a bi-criteria approximation for k-means with lower bounds. Notice that, unless we specify it, all these results hold for the more general case of non-uniform lower bounds even if we focused on uniform lower bounds in the introduction for simplicity.

These results presented in Chapter 2 are published in [17]. The work [17] is motivated by the master thesis *On Variants of Lower-Bounded Facility Location* [10] supervised by Melanie Schmidt, where we first develop the concept of weak lower bounds for the facility location problem and present an approximation algorithm for this problem. Furthermore we show how to modify a solution with weak lower bounds into a solution with 2-weak lower bounds while the cost only increases by a constant factor. The work [17] builds upon [10] but considers k-median and k-means with weak lower bounds. The approximation algorithms for k-median and k-means with weak lower bounds follow the same idea as in [10]. The modification of a solution with weak lower bounds into a solution with 2-weak lower bounds is similar to [10] but contains significant modifications

to work for k-means. The work [17] has been carried out together with Melanie Schmidt who also gave the initial idea for the concept of weak lower bounds. The reduction of lower-bounded generalized k-median to lower-bounded generalized facility location presented in Section 2.2 was developed by Melanie Schmidt. The proof that a solution with weak lower bounds can be transformed into a solution with 2-weak lower bounds (or $(1+\epsilon)$-weak lower bounds) was done by the author of this thesis as well as the development of the bi-criteria algorithm and the proof of its approximation factor.

1.3 Hierarchical Clustering

In the second part of this thesis we study hierarchical clustering, which is defined as follows.

Definition. *Let $(\mathcal{X}, d)$ be a metric space and $\mathcal{P} \subset \mathcal{X}$ a set of n points. A hierarchical clustering $\mathscr{H}$ is a sequence of clusterings $(\mathcal{H}_n, \dots, \mathcal{H}_1)$ of the set $\mathcal{P}$ such that $\mathcal{H}_k$ is a k-clustering and it satisfies the following property: for every $1 \leq k < k' \leq n$ and every cluster $A \in \mathcal{H}_{k'}$ there exists a cluster $B \in \mathcal{H}_k$ with $A \subset B$. In this case we call $\mathcal{H}_{k'}$ and $\mathcal{H}_k$ hierarchically compatible with each other.*

A hierarchical clustering is thus a sequence of clusterings where the clustering with k' clusters is a refinement of the clustering with $k < k'$ clusters. To evaluate the quality of a hierarchical clustering we fix a k-clustering objective and compare to the optimal cost on every level of granularity.

Definition. *Let* cost *be an objective function for k-clustering which is to be minimized and $\mathcal{O}_k$ an optimal k-clustering with respect to* cost*. We say that $\mathscr{H}$ has an approximation factor of α with respect to* cost *if $\mathscr{H}$ is an α-approximation on every level of granularity, i.e.,* $\mathsf{cost}(\mathcal{H}_k) \leq \alpha \cdot \mathsf{cost}(\mathcal{O}_k)$ *for all $1 \leq k \leq n$.*

We see that a small approximation factor α yields a strong guarantee for the hierarchical clustering on every level of granularity. In fact $\alpha = 1$ would imply that the hierarchical clustering is optimal on every level. However, this cannot be achieved on every clustering instance since there are not necessarily optimal clusterings $\mathcal{O}_n, \dots, \mathcal{O}_1$ with respect to cost which are hierarchically compatible. We can see in Figure 1.4 a clustering instance consisting of four points. For $\mathsf{cost} = \mathsf{diam}$ we have $\mathsf{diam}(\mathcal{O}_3) = 1$ and $\mathsf{diam}(\mathcal{O}_2) = 2$. However, $\mathcal{O}_3$ and $\mathcal{O}_2$ are not hierarchically compatible and it is easy to see that there exists no hierarchical clustering which has an approximation guarantee $< \frac{3}{2}$.

This yields the question whether there even exist α-approximations for constant α. This question was answered positively by Dasgupta and Long [39] and Charikar et al. [27]. They both present independently an algorithm that computes efficiently an 8-approximate hierarchical clustering with respect to the objectives $\mathsf{rad}, \mathsf{drad}$ and diam. It is known that under reasonable conditions both algorithms even compute the same clustering [37]. Mondal [66] claims in a recent work that there exists an efficient 6-approximation for the drad objective. However, we will later see that this algorithm does not achieve the claimed guarantee by presenting an instance where it computes only a 7-approximation. For the objectives med and mean Plaxton [70] proposed constant-factor approximations later on and Lin et al. [63] present a general framework that also leads to constant approximation guarantees for many objective functions including med and mean. Their framework can be applied to compute hierarchical clusterings for any

Figure 1.4: Here we see the optimal clusterings of size three and size two with respect to cost = diam. These two clusterings are not hierarchically compatible.

cost function which satisfies a certain nesting property, especially those of k-median and k-means. This yields a 20.71α-approximation for k-median and a 576β-approximation for k-means. Here $\alpha = 2.67059$ and $\beta = 5.912$ are the currently best approximation guarantees for k-median [32] and k-means [31]. These factors can be improved to 16α for k-median [36] and 32β for k-means [46].

There are still several questions arising: Are there algorithms which run in polynomial time and can improve upon these factors? The 8-approximation by [39, 27] is still the best known polynomial time algorithm for the objectives rad, drad and diam.

What is the best approximation guarantee that we can get in polynomial time under the assumption that P $\neq$ NP? So far we only have the hardness results from the flat k-center/k-diameter clustering which directly transform to the hierarchical setting. Unless P = NP there is no polynomial time α-approximation for hierarchical clustering for $\alpha < 2$ and the objectives drad, rad and diam. For drad, rad this is an immediate consequence of the reduction from dominating set presented by [52]. A similar reduction from clique cover yields the statement for diam. However, it is not clear whether it is even possible to construct hierarchical clusterings which have approximation guarantee $\alpha < 2$, which already yields the final question.

What is the best approximation guarantee if we are not restricted to polynomial time algorithms but are just interested in the best approximation factor that a hierarchical clustering can achieve? As we have seen earlier, Figure 1.4 shows an easy example where the approximation guarantee for the diameter is greater than one. Das and Kenyon-Mathieu [37] and Großwendt [46] present instances for diam and rad, drad, where no hierarchical clustering has an approximation guarantee smaller than 2. On the other hand Großwendt [46] proves an upper bound of 4 on the approximation guarantee of drad by using the framework of Lin et al. [63]. For the two other objectives diam, rad the best known upper bound results from the 8-approximation by [39, 27].

Recently other cost functions for hierarchical clustering were proposed, which do not compare to the optimal clustering on every level. Dasgupta [38] defines a new cost function for similarity measures and presents an $O(\alpha \log(n))$-approximation for the respective problem. This was later improved to $O(\alpha)$ independently by Charikar and Chatziafratis [26] and Cohen-Addad et al. [33]. Here α is the approximation guarantee of sparsest cut. However, Cohen-Addad et al. [33] prove that every hierarchical clustering is an $O(1)$-approximation to the corresponding cost function for dissimilarity measures when the dissimilarity measure is a metric. A cost function more suitable for Euclidean spaces was developed by Wang and Moseley [74]. They prove that a randomly generated hierarchical clustering performs poorly for this cost function and show that bisecting k-means computes an $O(1)$-approximation.

1.3.1 Complete Linkage

Aside from the theoretical results, there also exist greedy heuristics, which are more commonly used in applications. One very simple agglomerative algorithm is the following: starting from the clustering where every point is separate, it merges in every step the two clusters whose merge results in the smallest increase of the cost function. For the diam objective this algorithm is known as complete linkage. For convenience we refer to the respective algorithm for rad and drad as complete linkage as well. For the mean objective this is Ward's method [75]. Other popular greedy heuristics include single linkage and average linkage. Single linkage merges in every step the two clusters with smallest distance to each other and average linkage merges the two clusters with smallest average distance to each other. Here the distance of two clusters A, B is defined as $\min_{x\in A, y\in B} d(x, y)$ and the average distance is defined as $\frac{1}{|A||B|}\sum_{x\in A, y\in B} d(x, y)$.

Ackermann et al. [1] analyze the approximation guarantee of complete linkage in the Euclidean space. They show an approximation guarantee of $\Omega(\log(k))$ for the objectives drad, rad and diam assuming the dimension of the Euclidean space to be constant. Surprisingly the dependence on the dimension d of the Euclidean space varies with the choice of the objective function. For drad the approximation factor only depends linearly on d, for rad it depends exponentially on d and for diam it depends doubly exponentially on d. Later Großwendt and Röglin [44] improved the approximation guarantee to $O(1)$ under the assumption that d is constant. There are not many results regarding complete linkage in general metric spaces. Dasgupta and Long [39] show a lower bound of $O(\log(k))$ on the approximation guarantee with respect to diam. For Ward's method Großwendt et al. [45] show an approximation guarantee of 2 under the strong assumption that the optimal clusters are well separated. We analyze the approximation guarantee of complete linkage in general metric spaces. We show that the approximation guarantee of complete linkage is in $\Theta(k)$ for the drad. For diam we show a lower bound of $\Omega(k)$ and an upper bound of $O(k^{\ln(3)/\ln(2)})$ on the approximation factor. Our results show that within our standard definition of the approximation factor complete linkage does not perform better than single linkage, for which we show an upper bound of $O(k)$ on the approximation factor. This is surprising since single linkage is not designed to minimize the diam or drad objective. However, in the definition of the approximation factor we always consider the largest ratio between the cost of the clustering computed by complete linkage and the optimal clustering as we vary over all possible cluster sizes. In fact complete linkage for drad produces reasonable results for most of the cluster sizes especially when compared to single linkage. To go past the worst case definition of an approximation factor we therefore consider what approximation factor is achieved by both clustering methods *on average*. We define the *average approximation factor* of a hierarchical clustering as the average of all ratios between the cost of a k-clustering and that of an optimal k-clustering. We show that the average approximation factor of complete linkage for drad is in $O(\log(n))$ while the average approximation factor of single linkage is in $\Omega(n)$. Thus complete linkage for drad produces a hierarchical clustering that is better on average than that produced by single linkage.

1.3.2 Results

Price of hierarchy. We have seen an example where it is not possible to find a hierarchical clustering which is optimal on every level of granularity. Thus given some objective function cost we are interested in the smallest α for which there always exists

a hierarchical clustering which is an α-approximation with respect to cost.

Definition. *For* $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}, \mathsf{drad}\}$ *the* price of hierarchy $\rho_{\mathsf{cost}} \geq 1$ *is defined as follows.*

1. *For every instance* $(\mathcal{X}, \mathcal{P}, d)$*, there exists a hierarchical clustering* $\mathscr{H}$ *of* $\mathcal{P}$ *that is a* ρ_{cost}*-approximation with respect to* cost.
2. *For any* $\alpha < \rho_{\mathsf{cost}}$ *there exists an instance* $(\mathcal{X}, \mathcal{P}, d)$*, such that there is no hierarchical clustering of* $\mathcal{P}$ *that is an* α*-approximation with respect to* cost.

We compute the price of hierarchy for all three objectives. First we provide a better upper bound for the objectives $\mathsf{rad}, \mathsf{diam}$.

Theorem (3.1.2)**.** *For* $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}\}$ *we have* $\rho_{\mathsf{cost}} \leq 3 + 2\sqrt{2} \approx 5.828$.

Second we construct an instance where there is no hierarchical clustering with approximation factor $< 3 + 2\sqrt{2}$ for $\mathsf{rad}, \mathsf{diam}$ and approximation factor < 4 for drad.

Theorem (3.1.8)**.** *For* $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}\}$ *we have* $\rho_{\mathsf{cost}} \geq 3 + 2\sqrt{2}$ *and for* $\mathsf{cost} = \mathsf{drad}$ *we have* $\rho_{\mathsf{cost}} \geq 4$.

Notice that this implies that the price of hierarchy is exactly $3 + 2\sqrt{2}$ for $\mathsf{rad}, \mathsf{diam}$. For drad we can combine our result with the upper bound of 4 by [46] to see that the price of hierarchy is exactly 4. The upper bound of $3 + 2\sqrt{2}$ for the radius was also recently proved by Bock [22] in independent work.

Complete linkage. In the second part of this chapter we analyze the approximation guarantee of complete linkage and single linkage for the objectives drad and diam. Thus we can assume that the clustering instance $(\mathcal{X}, \mathcal{P}, d)$ satisfies $\mathcal{X} = \mathcal{P}$.

Theorem (3.2.4)**.** *Let* $\mathscr{H} = (\mathcal{H}_n, \ldots, \mathcal{H}_1)$ *be the hierarchical clustering computed by complete linkage on* $(\mathcal{P}, d)$ *with respect to* drad*. For all* $1 \leq k \leq n$ *the radius* $\mathsf{drad}(\mathcal{H}_k)$ *is upper bounded by* $O(k)\mathsf{drad}(\mathcal{O}_k)$*, where* $\mathcal{O}_k$ *is an optimal* k*-clustering with respect to* drad.

Theorem (3.2.12)**.** *Let* $\mathscr{H} = (\mathcal{H}_n, \ldots, \mathcal{H}_1)$ *be the hierarchical clustering computed by complete linkage on* $(\mathcal{P}, d)$ *with respect to* diam*. For all* $1 \leq k \leq n$ *the diameter* $\mathsf{diam}(\mathcal{H}_k)$ *is upper bounded by* $\lceil k^{\ln(3)/\ln(2)} \rceil \mathsf{diam}(\mathcal{O}_k)$*, where* $\mathcal{O}_k$ *is an optimal* k*-clustering with respect to* diam.

On the other hand we construct for both objectives instances where the cost of the k-clustering computed by complete linkage is by factor at least k worse than the cost of an optimal k-clustering.

Theorem (3.2.21)**.** *For every* $k \in \mathbb{N}$ *there exists an instance* $(V(P_k), d)$ *on which complete linkage, minimizing either* diam *or* drad*, computes a solution of diameter* k *or radius* $\frac{k}{2}$*, respectively, whereas the cost of an optimal solution is* 1.

Thus the performance guarantee of complete linkage in general metric spaces with respect to drad is in $\Theta(k)$. It remains an open question whether the analysis for diam can be improved to prove an approximation guarantee of $\Theta(k)$.

Finally we present an easy proof that complete linkage computes an $O(1)$-approximation in the Euclidean metric space. This result is a simplification of the proof presented by

Großwendt and Röglin [44] and implies slightly smaller approximation factors compared to [44].

Chapter 3 contains results from the two works [13] and [14].

The work [13] has been carried out together with Anna Großwendt, Heiko Röglin, Melanie Schmidt and Julian Wargalla. The paper is the result of many discussions between all authors and a dynamic developing process that all authors contributed to. Nevertheless, for the construction of the lower bound instance as well as the proof for the performance guarantee of single linkage, the main contribution is due to Anna Großwendt and Julian Wargalla. The proof of the upper bound on complete linkage for drad and diam is mainly due to the the author of this thesis, and the notion of the average approximation factor is solely due to the author of this thesis.

The work [14] has been carried out together with Heiko Röglin. Heiko Röglin suggested to study the price of hierarchy for different clustering objectives. The paper is the result of many discussions between both authors and a dynamic developing process that both authors contributed to. However the author of this thesis mainly developed the algorithm for the upper bound on the price of hierarchy for the objectives diam and rad as well as the clustering instance and the proof for the lower bound on the price of hierarchy for all three objectives.

Furthermore this chapter contains unpublished material. We give a simplified analysis of the last steps of complete linkage presented in [44] which also yields slightly better approximation factors.

1.4 Minimum-Error Triangulations

Given a set $\mathcal{S}$ on the Euclidean plane, a triangulation of $\mathcal{S}$ is a maximal set of straight line edges between points in $\mathcal{S}$ which do not intersect except at their endpoints. We define a triangle as the convex hull of three non-collinear points s, t, u and say that it is part of a triangulation of $\mathcal{S}$ if its vertices s, t, u are in $\mathcal{S}$, it does not contain further points from $\mathcal{S}$ and edges between the vertices s, t, u are part of the triangulation. The maybe most widely known triangulation is the Delaunay triangulation. A triangulation of $\mathcal{S}$ is called a Delaunay triangulation if for every triangle in the triangulation the circumcircle of the triangle does not contain points from $\mathcal{S}$ other than its vertices in its interior. If the points in $\mathcal{S}$ are in general position, i.e., no more than three points from $\mathcal{S}$ lie on a circle, such a Delaunay triangulation exists, is unique and can be computed in time $O(n \log(n))$ [40] where n is the number of points. Furthermore among all triangulations it maximizes the minimum angle of a triangle [40]. Usually one is interested in finding a triangulation which optimizes a certain objective function. A problem of this flavor is the minimum-weight triangulation problem, where the objective to be minimized is the sum of lengths of edges in the triangulation. There exist polynomial time algorithms which solve this problem on simple polygons [42, 56] as well as a constant factor approximation in the general case [59]. However, it has been a long standing open problem whether the minimum-weight triangulation can be computed efficiently until Mulzer and Rote [67] proved that this problem is NP-hard.

Now suppose that additionally to the triangulation points $\mathcal{S}$ there exists a function f assigning to every point in $\mathcal{S}$ a real value. Every triangulation D of the set $\mathcal{S}$ then yields a piece-wise linear function on the convex hull of $\mathcal{S}$, where the convex hull of a set of points is defined as follows.

Definition. *For a set* $M \subset \mathbb{R}^d$ *let* $\mathsf{conv}(M) = \{\sum_{i=1}^n \alpha_i x_i \mid \sum_{i=1}^n \alpha_i = 1, \alpha_i \geq 0, \{x_1, \ldots, x_n\} \subset M, n \in \mathbb{N}\}$ *denote the convex hull of* M*.*

For every triangle T with vertices s, t, u in the triangulation D we obtain a linear interpolation f_D of f as follows. For every point $v \in T$ there exist unique $\alpha, \beta, \gamma \geq 0$ with $\alpha + \beta + \gamma = 1$ and $v = \alpha s + \beta t + \gamma u$, so we define $f_D(v) = \alpha f(s) + \beta f(t) + \gamma f(u)$. Data-dependent triangulation problems, which were introduced by Dyn, Levin and Rippa in [41], are optimization problems related to properties of the function f_D. They considered the smoothness of the function f_D as an optimization criterion, i.e., the interpolation does not vary too much between neighboring vertices in the triangulation. Furthermore they suggest to study three-dimensional properties of the triangulation. Notice that we can move points from $\mathcal{S}$ into the three-dimensional space by appending the values given by the function f. Then every triangulation of $\mathcal{S}$ naturally transforms into a triangulation in the three dimensional space. Dyn, Levin and Rippa [41] propose to optimize properties of the three-dimensional triangulation, for example to maximize the minimum angle or to minimize the sum of edge lengths.

Finally they also introduced minimum-error criteria which are useful in the case that the function f is defined not only on $\mathcal{S}$ but on the convex hull of $\mathcal{S}$. A triangulation D is then considered good if the respective interpolation f_D represents the function f well. We call this problem the *minimum-error triangulation problem* and define it formally as follows.

An instance of the minimum-error triangulation problem consists of a set of *triangulation points* $\mathcal{S} \subset \mathbb{R}^2$ with *measurement values* given by the function $f\colon S \to \mathbb{R}$ and a set of *reference points* $\mathcal{R} \subset \mathsf{conv}(\mathcal{S})$ with *reference values* given by the function $h\colon \mathcal{R} \to \mathbb{R}$. For a triangulation D of $\mathcal{S}$ the *error* of the triangulation is defined as

$$\mathsf{Err}_D(\mathcal{R}) = \sum_{r \in \mathcal{R}} (f_D(r) - h(r))^2.$$

Thus the error measures how well the interpolation matches the reference values.

The minimum-error triangulation problem is of interest in the context of reconstructing the sea level in the last centuries which can help to understand the impact of climate change on the sea. For this purpose one can use sea level recordings starting at the late 17th century on tide gauge stations which are usually located at the sea shore. However, these tide gauge stations even today are sparsely distributed and not distributed uniformly over the globe. Since 1993 gridded altimeter data, obtained from measurements by satellite, is available and allows for a better reconstruction of the sea level. To provide a reconstruction of the sea level in years where satellite measurements are not available a common approach is to learn global base functions in years where satellite measurements are available [30]. Another approach by Olivieri and Spada [69] suggests to compute a triangulation of the sea surface with the tide gauge stations as triangulation points. Knowing the measurements at the tide gauge stations, this provides a reconstruction of the altimeter data on the whole sea surface: In every triangle one can reconstruct the altimeter data by linearly interpolating the measurements at the tide gauge stations that form the vertices of that triangle. Now the question arises how to find a reasonable triangulation of the sea surface. Olivieri and Spada [69] use the Delaunay triangulation to reconstruct the sea level of the Baltic Sea. We know that the Delaunay triangulation maximizes the minimum angle of a triangulation over all possible triangulations of the points [40] and therefore produces nicely shaped triangles. However, the Delaunay triangulation is known to be unique and independent of the altimeter data at the tide

gauge stations [40] so it does not allow for optimization over how well the triangulation represents the altimeter data obtained from satellite measurements. Therefore Nitzke et al. [68] propose to take the altimeter data from satellite measurements into account to compute a good triangulation of the sea surface. At a year where both tide gauge data as well as data from satellite measurements is available, we learn a minimum-error triangulation of the sea surface which minimizes the sum of the squared differences between the satellite altimeter data and its interpolation given by the triangulation and then use this triangulation to reconstruct the sea level in years where no satellite altimeter data is available. However, such triangulations might contain badly shaped triangles which only minimize the error at that specific year and do not generalize well to other years. Therefore Nitzke et al. [68] suggested to restrict the set of possible triangulations that one can choose from to minimize the error. They suggested to only consider k-order Delaunay triangulations. A k-order Delaunay triangulation introduced in [47] by Gudmundsson, Hammar and van Kreveld is a triangulation where every circumcircle of a triangle contains at most k points other than the vertices of the triangle in its interior. If $k = 0$ this gives us the definition of Delaunay triangulations and if k equals the number of triangulation points then every triangulation is a k-order Delaunay triangulation. The parameter k is a trade off between nicely shaped triangulations and triangulations with small error. Nitzke et al. [68] evaluate this approach for values $k \leq 3$ on the North Sea and show that their reconstruction outperforms the reconstruction obtained from the Delaunay triangulation in [69] for up to 19 years back in time. To find a triangulation of minimum error among all k-order Delaunay triangulations they use integer linear programming techniques.

In general there exist many heuristics to compute solutions for data-dependent triangulations [6, 24, 41, 73]. Most of these heuristics are based on a local search algorithm called Lawson's edge flip algorithm [58]. For a given triangulation Lawson's edge flip algorithm replaces one edge of the triangulation such that the given objective is improved and repeats this procedure.

For the minimum-weight triangulation problem there exist several algorithms running in FPT time [7, 23, 29, 42, 56] which can be adapted to the minimum-error triangulation problem [21]. In the case where the set of triangulations is restricted to 1-order Delaunay triangulations Gudmundsson, Hammar and van Kreveld [47] give a polynomial time algorithm to compute optimal solutions for some data-dependent triangulation problems which also yields a polynomial time algorithm to the minimum-error triangulation problem restricted to 1-order Delaunay triangulations. For k-order Delaunay triangulations Silveira and van Kreveld [71] present an algorithm for a class of data-dependent triangulation problems including the minimum-error triangulation problem, where the running time is not necessarily polynomial but depends exponentially on the number of connected components in the so called order-k Delaunay fixed-edge graph.

To compute minimum-error triangulations on a large set of points, for example to reconstruct the global sea surface, one may want to find efficient approximation algorithms for the minimum-error triangulation problem. Therefore the question arises whether this problem is NP-hard or even NP-hard to approximate within a certain factor. We know that the previously mentioned minimum-weight triangulation problem is NP-hard [67] as well as the surface-approximation problem [2] which is the problem of finding a piece-wise linear function of minimum complexity that approximates a given surface well.

1.4.1 Results

We show that the minimum-error triangulation problem is NP-hard to approximate. To show this we first prove the NP-hardness of the closely related *zero-error triangulation problem*. The zero-error triangulation problem asks for a triangulation D of $\mathcal{S}$ with $f_D(r) = h(r)$ for all $r \in \mathcal{R}$, or equivalently $\mathsf{Err}_D(\mathcal{R}) = 0$.

Theorem (4.3.1)**.** *The zero-error triangulation problem is* NP*-hard.*

We prove the NP-hardness of the zero-error triangulation problem via a reduction from planar 3SAT. Notice that the NP-hardness of this problem directly implies that the minimum-error triangulation problem likely cannot be approximated efficiently.

Corollary (4.3.2)**.** *The minimum-error triangulation problem cannot be approximated within any multiplicative factor in polynomial time unless* P = NP*.*

The following results, also contained in [11], are contributions of the other co-authors and are not discussed in detail in the main body of the dissertation. In the following we give a brief summary of these results.

For the sea surface reconstruction problem we propose to use the dynamic programming approach introduced by Silveira and van Kreveld [71] to compute a minimum-error triangulation with the restriction that the triangulation is a k-order Delaunay triangulation. The runtime of this algorithm depends exponentially on the number of connected components in the order-k Delaunay fixed-edge graph. For $k = 1$ we know that this graph is connected [47]. We analyze properties of this graph for $k = 2, 3$ and prove that for $k = 2$ every connected component consists of at least two vertices and present an instance whose order-k Delaunay fixed-edge graph is not connected. For $k \geq 3$ we present an instance whose order-k Delaunay fixed-edge graph consists of $\lfloor \frac{n}{6} \rfloor$ connected components, implying an exponential running time of the algorithm in the worst case. Here n is the number of triangulation points. However, the experiments on different projections of the tide gauge data set show that the algorithm is reasonably fast for data sets of medium size and $k \leq 7$ in practice. Finally we use the computed triangulation to reconstruct the sea surface at other years and evaluate its quality similarly to the approach by Nitzke et al. [68]. The experiments show that the reconstruction improves with larger k.

The work [11] which contains the NP-hardness proof presented in Chapter 4 has been carried out with Anne Driemel, Jan-Henrik Haunert, Herman Haverkort, Jürgen Kusche, Elmar Langetepe, Philip Mayer, Petra Mutzel and Heiko Röglin. Jan-Henrik Haunert suggested to study the hardness of the minimum-error triangulation problem. Anne Driemel and Herman Haverkort suggested the usage of the paraboloid for the definition of the reference and measurement values and constructed the wire gadget. Based on that idea the author of this thesis constructed the other gadgets and elaborated the NP-hardness proof including the proof of the properties of the gadgets and the final reduction from planar 3SAT. The properties of the bit were elaborated by Herman Haverkort.

Chapter 2

Achieving Anonymity via Weak Lower Bound Constraints for k-Median and k-Means

This chapter contains results from the work *Achieving Anonymity via Weak Lower Bound Constraints for k-median and k-means* [17] by Anna Arutyunova and Melanie Schmidt published in the proceedings of the *Symposium on Theoretical Aspects of Computer Science (STACS), 2021.* A full version of the paper is also available at *arXiv* [16].

Lower bounds were studied previously for the facility location problem and for k-center. Svitkina [72] presents the first constant factor approximation to facility location with uniform lower bounds. The approximation factor was later improved by Ahmadian and Swamy [5] to 82.6 and Li [61] gives the first constant factor approximation to facility location with non-uniform lower bounds. For k-center with lower bounds Aggarwal et al. [3] present a 2-approximation.

In this chapter we consider k-median and k-means with lower bounds. Given a set $\mathcal{P}$ we want to find a clustering of $\mathcal{P}$ where every cluster consists of at least B points while minimizing the k-median or k-means objective. We show that any solution which satisfies lower bounds but may open more than k centers can be transformed into a solution which satisfies lower bounds and opens at most k centers while the cost of the solution only increases by a constant factor. Furthermore we consider a relaxation of the lower bounds constraint which we call weak lower bounds and finally we present a bi-criteria approximation algorithm for both problems.

2.1 Preliminaries

Instead of considering k-median and k-means separately we define a generalization of both problems and consider this generalization in the following.

Definition 2.1.1. *An instance to the generalized k-median problem consists of a set $\mathcal{X}$ with distance function $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ and a set of points $\mathcal{P} \subset \mathcal{X}$. The distance function is symmetric and satisfies $d(x, y) = 0$ iff $x = y$. Furthermore there exists a constant $\alpha \geq 1$ such that d satisfies the α-relaxed triangle inequality, i.e., for all $x, y, z \in \mathcal{X}$ we have*

$$d(x, z) \leq \alpha(d(x, y) + d(y, z)).$$

A solution consists of a set $C \subset \mathcal{X}$ with $|C| \leq k$ and an assignment $\sigma\colon \mathcal{P} \to C$ with cost

$$\mathsf{med}(C, \sigma) = \sum_{x \in \mathcal{P}} d(x, \sigma(x)).$$

Notice that the *k-median problem* equals the generalized k-median problem with $\alpha = 1$. For the *k-means problem* in the Euclidean space we set d to be the squared Euclidean distance and $\mathcal{X} = \mathbb{R}^d$. However since $\mathcal{X}$ is assumed to be part of the input we restrict ourselves to the case where $\mathcal{X}$ is a finite set of size polynomial in $|\mathcal{P}|$. It is well-known that for any constant $\epsilon \in (0, 1)$ and for any $\mathcal{P} \subset \mathbb{R}^d$ we can compute a set $\mathcal{X}$ whose size is polynomial in $|\mathcal{P}|$ in polynomial time and that any γ-approximation yields a $(\gamma + \epsilon)$-approximation for the original problem [65]. Therefore we assume for the k-means problem that $\mathcal{X}$ is a finite set which is part of the input from now on. Observe that the squared Euclidean metric satisfies the relaxed triangle inequality for $\alpha = 2$ as the following easy computation shows.

Lemma 2.1.2. *For $x, y, z \in \mathbb{R}^d$ we have $\|x - z\|_2^2 \leq 2\|x - y\|_2^2 + 2\|y - z\|_2^2$.*

Proof. First notice that

$$0 \leq (\|x - y\|_2 - \|y - z\|_2)^2 = \|x - y\|_2^2 + \|y - z\|_2^2 - 2\|x - y\|_2\|y - z\|_2.$$

We use the triangle inequality for $\|\cdot\|_2$ to obtain

$$\|x - z\|_2^2 \leq (\|x - y\|_2 + \|y - z\|_2)^2 \leq 2\|x - y\|_2^2 + 2\|y - z\|_2^2. \qquad \square$$

Lower bounds have already been studied for facility location by [72, 61]. Similar to the generalized k-median problem we can consider a generalized version of the facility location problem, where the distance function only satisfies the relaxed triangle inequality.

Definition 2.1.3. *An instance to the generalized facility location problem consists of a set $\mathcal{X}$ with distance function $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$, a set of points $\mathcal{P} \subset \mathcal{X}$ and facility opening costs $f_x \geq 0$ for all $x \in \mathcal{X}$. The distance function is symmetric and satisfies $d(x, y) = 0$ iff $x = y$. Furthermore there exists a constant $\alpha \geq 1$ such that d satisfies the α-relaxed triangle inequality. A solution consists of a set $C \subset \mathcal{X}$ and an assignment $\sigma\colon \mathcal{P} \to C$ with cost*

$$\mathsf{fac}(C, \sigma) = \sum_{x \in \mathcal{P}} d(x, \sigma(x)) + \sum_{c \in C} f_c.$$

The generalized k-median problem with center costs equals the generalized facility location problem, with the only difference that the set C has to contain at most k centers.

Given a solution (C, σ) for generalized k-median or generalized facility location, we say that $\sigma^{-1}(c)$ is the cluster with center $c \in C$ and call $(\sigma^{-1}(c))_{c \in C}$ the clustering associated to (C, σ). In the *lower-bounded generalized k-median* and *lower-bounded generalized facility location* problem we demand that every center in the solution (C, σ) gets assigned a minimum number of points. For uniform lower bounds, the input additionally contains a number B and the solution must satisfy $|\sigma^{-1}(c)| \geq B$ for all $c \in C$. Non-uniform lower bounds are specified by a function $B : \mathcal{X} \to \mathbb{N}$ and the solution must satisfy $|\sigma^{-1}(c)| \geq B(c)$ for all $c \in C$.

Notice that Euclidean k-means with uniform lower bounds can be modeled by generalized k-median with uniform lower bounds since we can restrict the set $\mathcal{X}$ to have

size polynomial in $|\mathcal{P}|$ [65]. This does not work for Euclidean k-means with non-uniform lower bounds since by restricting the centers to be in $\mathcal{X}$ we may be forced to satisfy large lower bounds and therefore all solutions may have high cost compared to the cost of an optimal solution for the original problem. Therefore we only consider the variant of k-means with non-uniform lower bounds where the set of centers is already part of the input.

Clustering with lower bounds can be of interest when the set $\mathcal{P}$ contains data which needs to be protected. Enforcing lower bounds guarantees a certain level of anonymity. Therefore it would be sufficient to consider uniform lower bounds, however most of our results also hold for non-uniform lower bounds. Therefore most of the time we consider the general case of non-uniform lower bounds.

Our new problem variant called *weakly lower-bounded generalized k-median* is a relaxation of the lower bound constraint in the following way. Given lower bounds $B : \mathcal{X} \to \mathbb{N}$, the goal is to compute a set of at most k centers $C \subset \mathcal{X}$ and an assignment $\sigma \colon \mathcal{P} \to 2^C \setminus \{\emptyset\}$ such that the lower bound is satisfied, i.e., $|\{x \in \mathcal{P} \mid c \in \sigma(x)\}| \geq B(c)$ for all $c \in C$. Here 2^C denotes the power set of C. If a point is assigned to multiple centers the distance of the point to all assigned centers is paid by the solution. The total cost of a solution is given by

$$\mathsf{med}(C, \sigma) = \sum_{x \in \mathcal{P}} \sum_{c \in \sigma(x)} d(x, c).$$

If a solution of a weakly lower-bounded clustering problem satisfies that every point is assigned to at most b centers, then we say that the solution satisfies *b-weak lower bounds*. Weak lower bounds were already considered in [10] for the facility location problem.

2.2 Reducing Lower-Bounded k-Clustering to Lower-Bounded Facility Location

We observe that by using a known technique from the area of approximation algorithms for hierarchical clustering, we can turn an approximation algorithm for lower-bounded generalized facility location into an algorithm for lower-bounded generalized k-median. The technique is called *nesting*. Given two solutions (C_1, σ_1) and (C_2, σ_2) for the same clustering problem with different number of centers $k_1 > k_2$, nesting describes how to find a solution (C, σ) with k_2 centers which has a cost bounded by a constant times the costs of (C_1, σ_1) and (C_2, σ_2) and which is *hierarchically compatible* with (C_1, σ_1), i.e., the clusters in (C, σ) result from merging clusters in (C_1, σ_1). Suppose we consider the generalized k-median problem with lower bounds. Let (C_1, σ_1) be a solution which satisfies lower bounds but may violate the cardinality constraint of k for the number of cluster centers and (C_2, σ_2) be a solution which consists of at most k centers but may violate the lower bound constraint. The resulting solution (C, σ) has at most k centers and the clusters result from clusters that satisfy the lower bound – thus they satisfy the lower bound as well. For uniform lower bounds, the execution of this plan is very straightforward, for non-uniform lower bounds we have to be a bit more careful and adjust the nesting appropriately. Although the reduction is applicable to lower-bounded generalized k-median, this only helps to obtain constant-factor approximations for lower-bounded k-median because no approximation algorithms for generalized facility location with lower bounds are known for $\alpha > 1$.

More generally this approach can also be applied for generalized k-median under side constraints which are benign in the following sense: If a clustering satisfies the constraint, then the clustering resulting from merging two clusters is also feasible under the constraint. This is true for lower-bounded clustering since a cluster arising from merging two clusters with B points each definitely has at least B points, too. We call such constraints *mergeable constraints*. A slightly weaker mergability property holds for non-uniform lower bounds where the constraint depends on the center: If we merge two clusters that satisfy lower bounds $B(c_1)$ and $B(c_2)$ of their centers c_1 and c_2, then the merged cluster still satisfies the lower bounds of c_1 and c_2, so as long as the merged cluster uses one of these two centers, the lower bound is still satisfied.

For many clustering problems under side constraints, solving the version where the number of centers is at most k is much more difficult to tackle than solving its facility location variant where we allow arbitrary many centers and the facility opening costs are set to zero. For example, uniform capacitated facility location allows for a 3-approximation, while finding a constant-factor approximation for uniform capacitated k-median is a long-standing open problem. However, this is not the case for lower-bounded clustering because of the above described mergability property.

Roughly speaking, we show that for mergable constraints, we can turn a generalized k-median solution and a constrained facility location solution (with facility opening cost equal zero) into a constrained generalized k-median solution which does not cost much more. To do this, we borrow a concept from the area of *hierarchical* clustering which formalizes what it costs to merge clusters under a specific clustering objective.

Definition 2.2.1 (adapted from [63])**.** *A generalized facility location problem satisfies the (γ, δ)-nesting property for reals $\gamma, \delta \geq 0$ if for any input point set $\mathcal{P}$ and any two solutions $S_1 = (C_1, \sigma_1)$ and $S_2 = (C_2, \sigma_2)$ with $|C_1| > |C_2|$, a solution $S = (C, \sigma)$ can be computed such that*

- *S_1 and S are hierarchically compatible, i.e., for all $c \in C_1$ there exists a $c' \in C$ such that for all $x \in \mathcal{P}$ with $\sigma_1(x) = c$ it holds that $\sigma(x) = c'$,*
- *$\mathsf{fac}(C, \sigma) \leq \gamma \cdot \mathsf{fac}(C_1, \sigma_1) + \delta \cdot \mathsf{fac}(C_2, \sigma_2)$, and*
- *$|C| \leq |C_2|$.*

We call such a solution (γ, δ)-nested *with respect to S_1 and S_2.*

Lin et al. [63] show that the standard facility location/k-median cost function satisfies the $(2, 1)$-nesting property (also see Lemma 2.2.3 below). Combining this with the best-known constant-factor approximation for k-median [25], which achieves a $(2.67059 + \epsilon)$-approximation [32], and the 82.6-approximation for facility location with uniform lower bounds by Ahmadian and Swamy [4], the following lemma implies a $(167.87059 + \epsilon)$-approximation for k-median with uniform lower bounds.

Lemma 2.2.2. *Assume that we are given a generalized facility location problem with facility opening costs equal zero that satisfies the (γ, δ)-nesting property, a β-approximation algorithm for its generalized k-median variant and an α-approximation algorithm for the constrained generalized facility location variant under a mergeable constraint. Then there is a $(\gamma \cdot \alpha + \delta \cdot \beta)$-approximation algorithm for the generalized k-median problem under the same constraint.*

Proof. We compute two solutions: An α-approximate solution $S_1 = (C_1, \sigma_1)$ for the constrained generalized facility location variant and a β-approximate solution $S_2 = (C_2, \sigma_2)$ for the unconstrained generalized k-median problem. Remember that the opening cost for facilities equals zero.

By the nesting property, we get a solution $S = (C, \sigma)$ for the constrained generalized k-median problem which costs $\mathsf{med}(C, \sigma) \leq \gamma \cdot \mathsf{fac}(C_1, \sigma_1) + \delta \cdot \mathsf{med}(C_2, \sigma_2)$, that is hierarchically compatible with S_1 and satisfies that $|C| \leq |C_2| \leq k$. The unconstrained generalized k-median problem is a relaxation of the constrained generalized k-median problem because all we do is drop the constraint. The constrained facility location problem with facility opening cost zero arises from dropping the condition that $|C| \leq k$, so it is also a relaxation. Thus,

$$\begin{aligned}\mathsf{med}(C, \sigma) &\leq \gamma \cdot \mathsf{fac}(C_1, \sigma_1) + \delta \cdot \mathsf{med}(C_2, \sigma_2) \\ &\leq \gamma \cdot \alpha \cdot \mathsf{med}(\mathcal{O}_k) + \delta \cdot \beta \cdot \mathsf{med}(\mathcal{O}_k),\end{aligned}$$

where $\mathcal{O}_k$ is an optimal solution for the constrained generalized k-median problem. Since S is hierarchically compatible with S_1, we know that every cluster in S results from merging two clusters in S_1. Since S_1 satisfies the constraint and the constraint is mergeable, S also satisfies the constraint. □

Notice that Definition 2.2.1 does not require that the center set C is a subset of C_2. For (truly) mergable constraints like uniform lower bounds, this poses no problem because the constraint is not affected by the choice of center. However, for non-uniform lower bounds, we have to be a little more careful: We need that the merged cluster is assigned to a center whose lower bound is indeed satisfied. Definition 2.2.1 does not guarantee this. We thus prove the following slight generalization of the nesting step by Lin et al. [63] (Statement 1 only generalizes to the case of arbitrary α, but Statement 2 gives the generalization that we need for non-uniform lower bounds).

Lemma 2.2.3. *Let $S_1 = (C_1, \sigma_1)$ and $S_2 = (C_2, \sigma_2)$ be two solutions with $|C_1| > |C_2|$ for the generalized facility location problem with facility opening costs equal to zero. We can compute*

1. *a solution $S = (C_2', \sigma)$ with $C_2' \subseteq C_2$ that is $(\alpha + \alpha^2, \alpha^2)$-nested with respect to S_1 and S_2,*

2. *a solution $S = (C_1', \sigma')$ with $C_1' \subseteq C_1$ that is $(\alpha^3 + 2\alpha^2, \alpha^3 + \alpha^2)$-nested with respect to S_1 and S_2, and which satisfies that for all $c \in C_1'$ and for all $x \in \mathcal{P}$ with $\sigma_1(x) = c$, it holds that $\sigma'(x) = c$.*

Proof. We have two solutions $S_1 = (C_1, \sigma_1)$ and $S_2 = (C_2, \sigma_2)$ with $|C_1| > |C_2|$.

For all $c_i \in C_1$, let P_i be the set of all points assigned to $c_i \in C_1$ by σ_1, and for all $o_j \in C_2$, let O_j be the set of all points assigned to o_j by σ_2. First we create a solution $S = (C_2', \sigma)$ with $C_2' \subseteq C_2$. For all i, we assign every point $x \in P_i$ the center o_j which is closest to c_i, i.e., $\sigma(x) = \arg\min_{o \in C_2} d(c_i, o)$. By this choice we know that for any $x \in P_i$, $d(c_i, o_j) \leq d(c_i, \sigma_2(x))$. By two applications of the relaxed triangle inequality,

we get that

$$\begin{aligned}\sum_{x\in P_i} d(x,o_j) &\leq \sum_{x\in P_i} \alpha\cdot d(x,c_i) + \sum_{x\in P_i} \alpha\cdot d(c_i,o_j)\\ &\leq \alpha\cdot\sum_{x\in P_i} d(x,c_i) + \alpha\cdot\sum_{x\in P_i} d(c_i,\sigma_2(x))\\ &\leq \alpha\cdot\sum_{x\in P_i} d(x,c_i) + \alpha\cdot\sum_{x\in P_i} \alpha\cdot(d(c_i,x)+d(x,\sigma_2(x)))\\ &= (\alpha+\alpha^2)\cdot\sum_{x\in P_i} d(x,\sigma_1(x)) + \alpha^2\cdot\sum_{x\in P_i} d(x,\sigma_2(x)).\end{aligned}$$

Adding the cost of all clusters yields the statement.

Now we convert S into a solution (C_1',σ') with $C_1'\subset C_1$ at the cost of an increase in the nesting factors. Let i be fixed. So far, we have reassigned the points in P_i to the center o_j in C_2 closest to c_i. Now among all $c\in C_1$ for which o_j was the closest center, we choose a center that is closest to o_j and reassign the points there. More formally let $D_j=\{\sigma_1(x)\mid x\in\mathcal{P},\sigma(x)=o_j\}$ we define $\sigma'(x)=\arg\min_{c\in D_j} d(o_j,c)$ for all points $x\in P_i$. The points are now assigned only to points in C_1. We know that all points assigned to o_j by σ are (re)assigned to $\arg\min_{c\in D_j} d(o_j,c)$. And because we only reassign a new center to the clustering induced by (C_2',σ), we know that it still has at most $|C_2'|\leq|C_2|$ many clusters. The cost is bounded by

$$\begin{aligned}\sum_{x\in P_i} d(x,\sigma'(x)) &\leq \sum_{x\in P_i} \alpha\cdot d(x,o_j) + \alpha\cdot d(o_j,\sigma'(x))\\ &= \sum_{x\in P_i} \alpha\cdot d(x,o_j) + \alpha\cdot d(o_j,c_i)\\ &\leq \sum_{x\in P_i} \alpha^2\cdot d(x,c_i) + (\alpha^2+\alpha)\cdot d(o_j,c_i)\\ &\leq \sum_{x\in P_i} \alpha^2 d(x,c_i) + (\alpha^2+\alpha)\sum_{x\in P_i}\alpha\cdot(d(c_i,x)+d(x,\sigma_2(x)))\\ &= (\alpha^3+2\alpha^2)\cdot\sum_{x\in P_i} d(x,\sigma_1(x)) + (\alpha^3+\alpha^2)\cdot\sum_{x\in P_i} d(x,\sigma_2(x)).\end{aligned}$$

□

Statement 2 of Lemma 2.2.3 guarantees a solution where the centers are a subset of the centers in solution S_1, and the assignment ensures that points that were previously assigned to the chosen centers C_1' are still assigned to their previous center. This has the following benefit: if the mergeability of the constraint depends on the center as for non-uniform lower bounds, we still satisfy the constraint. Indeed, for all $c\in C_1'$ we now know that all points previously assigned to c are still assigned to c, then this means that if the lower bound for c was satisfied by S_1, then it is also satisfied for S. We obtain the following result for k-median with lower bounds as an immediate consequence of Lemma 2.2.3.

Corollary 2.2.4. *Suppose there exists a λ-approximation algorithm for k-median and a μ-approximation algorithm for facility location with uniform or non-uniform lower bounds. Then there exists a $(2\mu+\lambda)$-approximation for k-median with uniform lower bounds and a $(3\mu+2\lambda)$-approximation for k-median with non-uniform lower bounds.*

Thus, we plug in the the $O(1)$-approximation for facility location with non-uniform lower bounds by Li [61] as S_1 and the already mentioned $(2.67059+\epsilon)$-approximation [32] for k-median as S_2 and get an $O(1)$-approximation for k-median with non-uniform lower bounds.

2.3 Generalized k-Median with Weak Lower Bounds

Now we consider a relaxed version of generalized k-median with lower bounds where points in $\mathcal{P}$ can be assigned multiple times. This relaxation does make sense since we have lower bounds on the centers, so it can be more valuable to assign points to multiple centers to satisfy the lower bounds instead of closing the respective centers.

For standard k-median/k-means with weak lower bounds we give an algorithm that computes a $(6.5+\epsilon)$-approximation and an $O(1)$-approximation respectively. Furthermore we show that a solution to generalized k-median with weak lower bounds can be transformed into a solution to generalized k-median with *2-weak lower bounds* in polynomial time. We show that this transformation increases the cost only by a factor of $\alpha(\alpha+1)$. We combine this with the approximation algorithm for standard k-median/k-means with weak lower bounds and obtain an approximation algorithm for standard k-median/k-means with 2-weak lower bounds. If we allow fractional assignments we show how to obtain a solution which assigns every point by an amount of at most $1+\epsilon$ for arbitrary $\epsilon \in (0,1)$, losing $\lceil \frac{1}{\epsilon} \rceil \alpha(\alpha+1)+1$ in the approximation factor.

The results in this section hold for the more general case of non-uniform lower bounds. Therefore we only consider non-uniform lower bounds from now on.

2.3.1 Computing a Solution

This paragraph mainly follows the ideas in [10], where it is described how one can obtain an approximation algorithm for weakly lower-bounded facility location. We adapt the approach in [10] for weakly lower-bounded generalized k-median.

To approximate generalized k-median with weak non-uniform lower bounds, we reduce this problem to generalized k-median with center costs. Remember that this problem is a combination of generalized k-median and generalized facility location where we are only allowed to choose at most k centers and we have to pay a certain cost for opening centers.

The reduction that we use works by introducing a center cost of

$$f_c = \sum_{p \in D_c} d(p,c) \tag{2.1}$$

for every point $c \in \mathcal{X}$. This cost is paid if c becomes a center. Here D_c is the set consisting of the $B(c)$ nearest points in $\mathcal{P}$ to c. The idea for this reduction is adapted from the bi-criteria algorithm for lower-bounded facility location presented by Guha, Meyerson and Munagala [49] and Karger, Minkoff [55].

Note that for a center c in a feasible solution (C,σ) to generalized k-median with weak lower bounds, the term $\sum_{p \in D_c} d(p,c)$ is a lower bound on the assignment cost caused by c. This leads to the following lemma.

Lemma 2.3.1. *Let $\mathcal{O}'$ be an optimal solution to the generalized k-median problem with center costs and $\mathcal{O} = (O,\sigma)$ be an optimal solution to generalized k-median with weak lower bounds. It holds that* $\mathsf{fac}(\mathcal{O}') \leq 2\mathsf{med}(\mathcal{O})$.

Proof. For $p \in \mathcal{P}$ let $c_p = \operatorname{argmin}\{d(p,c) \mid c \in \sigma(p)\}$ be the closest center to which p is assigned in $\mathcal{O}$. We define $\sigma'(p) = c_p$ for all $p \in \mathcal{P}$ and obtain a feasible solution (O,σ')

to the generalized k-median problem with center costs. Furthermore we have

$$\begin{aligned}\mathsf{fac}(\mathcal{O}') \leq \mathsf{fac}(O, \sigma') &= \sum_{c \in O} f_c + \sum_{p \in \mathcal{P}} d(p, \sigma'(p)) \\ &= \sum_{c \in O} \sum_{p \in D_c} d(p, c) + \sum_{p \in \mathcal{P}} d(p, \sigma'(p)) \\ &\leq 2 \sum_{p \in \mathcal{P}} \sum_{c \in \sigma(p)} d(p, c) \\ &= 2\mathsf{med}(\mathcal{O}).\end{aligned}$$

The second inequality follows from the fact that $\sum_{c \in O} \sum_{p \in D_c} d(p, c)$ and $\sum_{p \in \mathcal{P}} d(p, \sigma'(p))$ are both lower bounds on the assignment cost of $\mathcal{O}$. □

Let (C, σ) be a solution for the generalized k-median problem with center costs. To turn it into a solution for generalized k-median with weak lower bounds we have to modify the assignment. Let $c \in C$ and $n_c = |\sigma^{-1}(c)|$. We additionally assign $m_c = \max\{0, B(c) - n_c\}$ points to c to satisfy the lower bound. Let $S_c \subset D_c$ be the set of points in D_c which are not assigned to c. We choose m_c points from S_c and assign them to c. This is feasible since we are allowed to assign points multiple times. Let (C, σ') be the corresponding solution.

Lemma 2.3.2. *It holds that* $\mathsf{med}(C, \sigma') \leq \mathsf{fac}(C, \sigma)$.

Proof. The additional assignment cost for each center $c \in C$ can be upper bounded by $\sum_{p \in D_c} d(p, c)$. We obtain

$$\begin{aligned}\mathsf{med}(C, \sigma') \leq &\sum_{c \in C} \sum_{p \in D_c} d(p, c) + \sum_{p \in \mathcal{P}} d(p, \sigma(p)) \\ =&\mathsf{fac}(C, \sigma).\end{aligned}$$

□

Lemma 2.3.1 and Lemma 2.3.2 imply the following corollary.

Corollary 2.3.3. *Given a γ-approximation for the generalized k-median problem with center costs, we get a 2γ-approximation for the generalized k-median problem with weak lower bounds in polynomial time.*

We use the $(3.25 + \epsilon)$-approximation for the k-median problem with center costs [28], which results in a $(6.5 + \epsilon)$-approximation for k-median with weak lower bounds. For k-means, we use the algorithm by Jain and Vazirani [53] which was originally designed for k-median. However, as outlined in the journal version [53], it can be used for k-means, and also for k-median with center costs. The two extensions are not conflicting and can both be applied to obtain an $O(1)$-approximation for the k-means problem with center costs.

2.3.2 Reducing the Number of Assignments per Point to 2

We see that the solution for standard k-median/k-means with weak lower bounds computed above can assign a point to all centers in the worst case. The number of assigned centers per point cannot be bounded by a constant. This may not be desirable in the

context of publishing anonymized representatives since the distortion of the original data set is not bounded.

However, we show that any solution to the generalized k-median problem with weak lower bounds can be transformed into a solution assigning every point at most twice. This increases the cost by a factor of $\alpha(\alpha+1)$. Recall that α is the constant appearing in the relaxed triangle inequality. This leads to the following theorem.

Theorem 2.3.4. *Given a solution (C,σ) to generalized k-median with weak lower bounds, we can compute a solution $(\widetilde{C},\widetilde{\sigma})$ to generalized k-median with 2-weak lower bounds (assigning every point at most twice) in polynomial time such that* $\mathsf{med}(\widetilde{C},\widetilde{\sigma}) \leq \alpha(\alpha+1)\,\mathsf{med}(C,\sigma)$.

A similar statement is already known for weakly lower-bounded facility location in [10], where the authors show that we can obtain a solution with 2-weak lower bounds for facility location while the cost of the respective solution increases by a factor of 2. While our proof is inspired by their approach it differs in the reassignment process to construct a solution with 2-weak lower bounds. This is necessary to achieve that the cost only increases by a constant factor, especially for distances which are not determined by a metric.

Reassignment process. We start by setting $\widetilde{C} = C$ and $\widetilde{\sigma} = \sigma$ and modify both $\widetilde{C}$ and $\widetilde{\sigma}$ until we obtain a feasible solution to generalized k-median with 2-weak lower bounds. During the process, the centers in $\widetilde{C}$ are called *currently open*, and when a center is deleted from $\widetilde{C}$, we say it is *closed*. The centers are processed in an arbitrary but fixed order, i.e., we assume that $C = \{c_1,\dots,c_{k'}\}$ for some $k' \leq k$ and process them in order $c_1,\dots,c_{k'}$. We say that c_i is *smaller* than c_j if $i<j$.

Let $c = c_i$ be the currently processed center. By P_c, we denote the set of points assigned to c under $\widetilde{\sigma}$. We divide P_c into three sets $P_c^1 = \{q \in P_c \mid |\widetilde{\sigma}(q)| = 1\}$, $P_c^2 = \{q \in P_c \mid |\widetilde{\sigma}(q)| = 2\}$ and $P_c^3 = \{q \in P_c \mid |\widetilde{\sigma}(q)| \geq 3\}$. Furthermore with $C(P_c^3)$ we denote all centers which are connected to at least one point in P_c^3 under $\widetilde{\sigma}$.

If P_c^3 is empty, we are done and proceed with the next center in $\widetilde{C}$. Otherwise we need to empty P_c^3. Observe that points in P_c^3 are assigned to multiple centers, so if we delete the connection between one of these points and c, the point is still served by some other center. However, doing so may violate the lower bound at c. So we have to replace this connection.

As long as P_c^3 is non-empty, we do the following. We pick a center $d = \min C(P_c^3)\backslash\{c\}$ and a point $x \in P_c^3$ connected to d. We want to assign a point y from P_d^1 to c to free x. For technical reasons, we restrict the choice of y: We exclude all points from the subset $\overline{P_d^1} := \{q \in P_d^1 \mid |\sigma(q)| \geq 3 \text{ and } \sigma(q) \cap \{c_1,\dots,c_{i-1}\} \cap \widetilde{C} \neq \emptyset\}$, i.e., all points which were assigned to at least 3 centers under the initial assignment σ, and where one of these at least 3 centers is still open *and* smaller than c.

If $P_d^1 \backslash \overline{P_d^1}$ is non-empty, we pick a point $y \in P_d^1 \backslash \overline{P_d^1}$ arbitrarily. We set $\widetilde{\sigma}(y) = \{d,c\}$ and $\widetilde{\sigma}(x) = \widetilde{\sigma}(x)\backslash\{c\}$. So x is no longer connected to c, but to satisfy the lower bound at c we replace x by y (Figure 2.1).

If $P_d^1 \backslash \overline{P_d^1}$ is empty, our replacement plan does not work. Instead, we close d. This means that x is now assigned to one center less, and, if this happens repeatedly, x will at some point no longer be in P_c^3. Since we close d, all points in P_d^1 have to be reassigned because they are only connected to d. For each $q \in P_d^1$, we reassign q to the smallest

Algorithm 1: Reducing the number of assigned centers per point to two

define an ordering on the centers $c_1 \leq c_2 \ldots \leq c_{k'}$
set $\widetilde{C} := C$ and $\widetilde{\sigma} := \sigma$
for all $c \in C$
  $P_c := \{q \in \mathcal{P} \mid c \in \widetilde{\sigma}(q)\}$
  $P_c^3 := \{q \in P_c \mid |\widetilde{\sigma}(q)| \geq 3\}$, $P_c^i := \{q \in P_c \mid |\widetilde{\sigma}(q)| = i\}$ for $i = 1, 2$
  $C(P_c^3) := \bigcup_{q \in P_c^3} \widetilde{\sigma}(q)$
for $i = 1$ **to** l **do**
  while $P_{c_i}^3 \neq \emptyset$ **do**
    $d = \min C(P_{c_i}^3) \backslash \{c_i\}$
    $\overline{P_d^1} = \{q \in P_d^1 \mid |\sigma(q)| \geq 3 \text{ and } \sigma(q) \cap \{c_1, \ldots, c_{i-1}\} \cap \widetilde{C} \neq \emptyset\}\}$
    if $P_d^1 \backslash \overline{P_d^1} = \emptyset$ **then**
      for all $q \in P_d^1$
        let $e = \min(\sigma(q) \cap \widetilde{C})$
        set $\widetilde{\sigma}(q) = \{e\}$
      delete d from $\widetilde{C}$ and all connections to d in $\widetilde{\sigma}$
    else
      pick $x \in P_{c_i}^3$ connected to d and $y \in P_d^1 \backslash \overline{P_d^1}$
      set $\widetilde{\sigma}(x) = \widetilde{\sigma}(x) \backslash \{c_i\}$, $\widetilde{\sigma}(y) = \{c_i, d\}$

currently open center in $\sigma(q)$. Notice that such a center exists and is smaller than c because $P_d^1 = \overline{P_d^1}$ and for every $q \in \overline{P_d^1}$, there is at least one center in $\sigma(q) \cap \widetilde{C}$ which is smaller than c.

The entire process is described in Algorithm 1. It satisfies the following invariants.

Lemma 2.3.5. *Algorithm 1 computes a feasible solution $(\widetilde{C}, \widetilde{\sigma})$ to generalized k-median with 2-weak lower bounds. Furthermore the following properties hold during all steps of the algorithm.*

1. *The algorithm never establishes connections for points currently assigned more than once.*
2. *For any center $c \in C$, P_c does not change before c is processed or closed.*
3. *If a connection between $x \in \mathcal{P}$ and the currently processed center $c \in \widetilde{C}$ is deleted by the algorithm, we have from this time on $x \notin P_c^3$ until termination. Moreover P_c^3 remains empty after c is processed.*

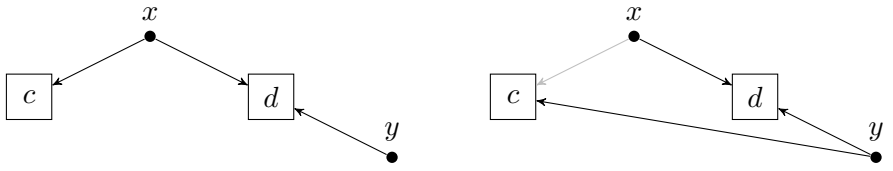

Figure 2.1: Connection between $x \in P_c^3$ and c is deleted. A point $y \in P_d^1$ replaces x.

4. *While the algorithm processes $c \in C$ we always have $c < \min C(P_c^3)\backslash\{c\}$. Moreover all currently open centers which are smaller than c remain open until termination.*

5. *If the algorithm establishes a new connection in Line 14 or Line 18 it remains until termination.*

Proof. The process terminates: For every iteration of the while loop starting in Line 8, either a point is deleted from $P_{c_i}^3$ or there is at least one point $x \in P_{c_i}^3$ for which $|\widetilde{\sigma}(x)|$ is reduced by one. Furthermore $|\widetilde{\sigma}(x)|$ does never increase for any $x \in P_{c_i}^3$.

The final solution satisfies lower bounds: Every time we delete a connection between a point and a center it either happens because the center is closed or we replace this connection by assigning a new point to it. So the lower bounds are satisfied at all open centers.

All points stay connected to a center: Assume that the algorithm deletes the connection between a point p and the center d it is exclusively assigned to. This only happens if at this time d is closed by the algorithm. Then p is assigned to another center as defined in Line 14.

We conclude that the solution is feasible.

Property 1: The algorithm establishes connections in Line 14 and Line 18 which always involve a point currently assigned once.

Property 2: Let $c \in C$. Connections are only changed for the center that is currently processed or for a smaller center which has been processed already. Thus, the algorithm does not add or delete any connections involving c before c is processed or closed.

Property 3: Assume that after the connection between $x \in P_c^3$ and c is deleted by the algorithm, x is again part of P_c^3. That would require that the algorithm establishes a new connection for a point which is connected more than once, which does not happen by Property 1. For the same reason P_c^3 remains empty after c is processed by the algorithm.

Property 4: Assume c is currently processed by the algorithm and $d = \min C(P_c^3)\backslash\{c\}$. We know that at this time P_d^3 is non-empty, which is by Property 3 only possible if d is processed after c. Thus we have $c < d$. This also means that centers can only be closed by the algorithm if they are not processed so far.

Property 5: If a connection is deleted, the respective point is either connected to more than two centers or to a center which is closed at this time. A connection in Line 14 or Line 18 is established by the algorithm between a point which is at this time assigned exactly once and a center which is already processed or currently processed by the algorithm. Thus the point is from this time on never assigned to more than two centers and the center remains open until termination by Property 4. So the necessary conditions for a deletion of this connection are never fulfilled. □

We now want to bound the cost of new connections created by the algorithm by the cost of the original solution. Notice that only Line 18 generates new connections, Line 14 re-establishes connections that were originally present. So let N_c be the set of all points newly assigned to c by the algorithm in Line 18 while center c is processed. For $y \in N_c$ let d_y be the respective center in Line 9 of Algorithm 1 and x_y the point in Line 17 contained in P_c^3 and connected to d_y.

Using the α-relaxed triangle inequality, we obtain the following upper bound.

$$\begin{aligned} d(y,c) &\leq \alpha(d(y,x_y)+d(x_y,c)) \leq \alpha\Big(\alpha\Big(d(y,d_y)+d(d_y,x_y)\Big)+d(x_y,c)\Big) \\ &= \alpha^2\Big(d(y,d_y)+d(d_y,x_y)\Big)+\alpha d(x_y,c). \end{aligned} \tag{2.2}$$

We can apply (2.2) to all $c \in \widetilde{C}$ and all $y \in N_c$. This yields the following upper bound on the cost of the final solution $(\widetilde{C}, \widetilde{\sigma})$.

$$\begin{aligned} \mathsf{med}(\widetilde{C},\widetilde{\sigma}) &= \sum_{c\in\widetilde{C}} \sum_{\substack{y\in\mathcal{P}:\\ c\in\widetilde{\sigma}(y)}} d(y,c) = \sum_{c\in\widetilde{C}} \Big(\sum_{y\in P_c\setminus N_c} d(y,c) + \sum_{y\in N_c} d(y,c)\Big) \\ &\leq \sum_{c\in\widetilde{C}} \Big(\sum_{y\in P_c\setminus N_c} d(y,c) + \sum_{y\in N_c} \alpha^2(d(y,d_y)+d(d_y,x_y)) + \alpha d(x_y,c)\Big). \end{aligned} \tag{2.3}$$

Expression (2.3) is what we want to pay for. We show in Observation 2.3.6 below that all involved distances contribute to the original cost as well. So in principle, we can bound each summand by a term in the original cost. But what we need to do is to bound the number of times that each term in the original cost gets charged. To organize the counting, we count how many times a specific tuple of a point z and a center f occurs as $d(z,f)$ in (2.3). Since it is important at which position a tuple appears, we give names to the different occurrences (also see Figure 2.2).

We say that that a tuple appears as a tuple of Type 0 if it appears as $d(y,c)$ in (2.3), as tuple of Type 1 if it appears as $d(x_y,c)$, and as tuple of Type 2 if it appears as $d(y,d_y)$ or $d(d_y,x_y)$. We distinguish the latter type further by calling a tuple occurring as $d(y,d_y)$ a tuple of Type 2.1 and a tuple occurring as $d(x_y,d_y)$ a tuple of Type 2.2. We say that $(y,d_y),(d_y,x_y)$ and (x_y,c) *contribute* to the cost of (y,c), where by the *cost* of (y,c) we mean the upper bound on $d(y,c)$ in (2.2) which we want to pay for.

Observation 2.3.6. *If a tuple (z,f), $z \in \mathcal{P}, f \in C$, occurs as Type 0, 1 or 2, then $f \in \sigma(z)$, so in particular, $d(z,f)$ occurs as a term in the cost of the original solution.*

Proof. For a center c the set $P_c\setminus N_c$ consists of points which are assigned to c by the initial assignment σ or assigned to c while c is not processed by the algorithm. The latter can only happen if a connection is reestablished in Line 14 which requires that the connection was already present in (C,σ). So Type 0 tuples satisfy the statement.

For Type 1 and 2 tuples, consider $y \in N_c$ for some center c and the respective tuples $(x_y,c),(y,d_y),(x_y,d_y)$. Notice that both y and x_y are connected to d_y the step before y is assigned to c. By Property 4 of Lemma 2.3.5 we have $c < d_y$. Thus we know by Property 2 of Lemma 2.3.5 that P_{d_y} is not changed by the algorithm at least until y is assigned to c. So $d_y \in \sigma(y)$ and $d_y \in \sigma(x_y)$ which proves that Type 2 tuples satisfy the statement. Moreover it holds that $c \in \sigma(x_y)$ since there is a time where $x_y \in P_c^3$. This can, by Property 1 of Lemma 2.3.5, only happen if the connection between x_y and c is already part of (C,σ). Thus, Type 1 tuples satisfy the statement. □

As indicated above, a tuple (z,f) can contribute to the cost of multiple tuples. Notice that a tuple occurs at most once as a tuple of Type 0 in (2.3). To bound the cost of $(\widetilde{C},\widetilde{\sigma})$ we bound the number of times a tuple appears as Type 1 or Type 2 tuple in (2.3).

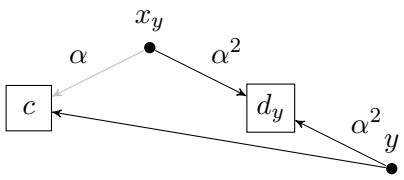

Figure 2.2: Bounding the distance between y and c. The respective distances appear with a factor of α or α^2. Tuple (x_y, c) is of Type 1 and $(x_y, d_y), (y, d_y)$ are of Type 2.

Lemma 2.3.7. *For all $z \in \mathcal{P}, f \in C$, the tuple (z, f) can appear in (2.3) at most once as a tuple of Type 1 and at most once as a tuple of Type 2.*

Proof. In the following, the tuple whose cost the tuple (z, f) contributes to will always be named (y, c), and we denote the time at which y is newly assigned to c by t.

Type 1: Assume (z, f) contributes to the cost of (y, c) as a tuple of Type 1. Then $f = c$. Notice that at the time step before t we must have $z \in P_c^3$ and afterwards, z is never again contained in P_c^3 by Property 3 of Lemma 2.3.5. Thus the pair (z, c) can never again be responsible for any reassignment to c, i.e., $(z, c) = (z, f)$ does not contribute to any further cost as a tuple of Type 1.

Type 2.1: Assume that (z, f) contributes to the cost of (y, c) as a tuple of Type 2.1. Then $z = y$. At the time step before t, we have $y \in P_f^1$, $f \in C(P_c^3)$, and at time t, we have $y \in P_c^2 \cap P_f^2$. By Property 5 of Lemma 2.3.5, newly established connections are never deleted, so after time t, it always holds that $y \in P_c$. So even if y is in P_f at a later time, it cannot be in P_f^1 since it is also connected to c. So $(y, f) = (z, f)$ does not contribute to any further cost as tuple of Type 2.1. Furthermore by Property 1 of Lemma 2.3.5 we know that y is always assigned to fewer than three centers after t which means that (y, f) does not contribute as tuple of Type 2.2 to the cost of any connection established by the algorithm after t either.

Type 2.2: Finally we consider the case where (z, f) contributes to the cost of (y, c) as a tuple of Type 2.2. At time t, the algorithm processes c. By the way the algorithm chooses f and z, we know that $z \in P_c^3$ (at the beginning of the process, i.e., before t) and $f = \min C(P_c^3) \backslash \{c\}$. After t, Property 3 of Lemma 2.3.5 implies $z \notin P_c^3$, which means that as a tuple of Type 2.2, it can never again contribute to the cost of any tuple containing c. Assume instead that it contributes (as Type 2.2) to the cost of a tuple (y', c') for a center $c' \neq c$, and some point $y' \in \mathcal{P}$. This is supposed to happen after t, so y' is newly assigned to c' at some time $t' > t$. Before c' is processed, we must always have $z \in P_{c'}^3$ by Property 1 and 2 of Lemma 2.3.5. So in particular, at time $t < t'$ we have $c' \in C(P_c^3) \backslash \{c\}$. Moreover we know that at some time while c' is processed by the algorithm we have $f = \min C(P_{c'}^3) \backslash \{c'\}$. Using Property 4 of Lemma 2.3.5 we conclude that $c' < f$. Which is a contradiction since the algorithm chose f and not c' at time t, i.e., $f = \min C(P_c^3) \backslash \{c\}$ must hold. Thus, (z, f) cannot contribute to the cost of (y', c') as a tuple of Type 2.2.

It is left to show that (z, f) cannot contribute to the cost of any (y', c') as a tuple of Type 2.1 at some time $t' > t$. For a contribution as Type 2.1, we would have $z = y'$ and $y' \in P_f^1$. We show that in this case y' is in fact contained in $\overline{P_f^1}$. Remember that at time t we have $y' = z \in P_c^3$ and that this only happens if $|\sigma(y')| \geq 3$ by Property 1 of Lemma 2.3.5. Moreover c is still open by Property 4 of Lemma 2.3.5 and is smaller than c'. Thus $c \in \sigma(y') \cap \{e \mid e < c'\} \cap \widetilde{C}$, which proves $y' \in \overline{P_f^1}$. Therefore the algorithm does

not assign y' to c' (see Lines 11-15) and (z, f) does not contribute as tuple of Type 2.1 to the cost of any connection established by the algorithm after t. □

We now know that a tuple only appears at most once as any of the three tuple types. For the final counting, we define $T0$, $T1$ and $T2$ as the sets of all tuples of Type 0, 1 and 2, respectively. We could already prove a bound on the cost now, but to make it slightly smaller and prove Theorem 2.3.4, we need one final statement.

Lemma 2.3.8. *The set $T0 \cap T1 \cap T2$ is empty.*

Proof. Let $(z, f) \in T0 \cap T1 \cap T2$. Since (z, f) is of Type 0, the point z must be connected to f in the final assignment $\widetilde{\sigma}$. We distinguish whether the connection between z and f was deleted at some point by the algorithm or not. If it is not deleted, (z, f) cannot be of Type 1 since this would require that z is temporarily not assigned to f. Otherwise the connection between z and f was deleted while f was processed and later reestablished by the algorithm in Line 14.

By assumption the tuple is also of Type 2. Assume it is of Type 2.1 and contributes to the cost of a tuple (y, c) with $z = y$. We know that $c < f$ by Property 4 of Lemma 2.3.5. Consider the time when z is newly assigned to c. The step before we have $z \in P_f^1$. On the other hand while f is processed we have $z \in P_f^3$ in contradiction to Property 1 of Lemma 2.3.5.

Assume finally that (z, f) is of Type 2.2 and contributes to the cost of a tuple (y, c). Again we have $c < f$. Consider the time y is newly assigned to c. The step before we have $z \in P_c^3$ and, by Property 1 and 2 of Lemma 2.3.5, also $z \in P_f^3$. At the time the connection between z and f is reestablished by the algorithm, both centers are contained in $\sigma(z) \cap \widetilde{C}$. This is a contradiction to $c < f = \min(\sigma(z) \cap \widetilde{C})$. This completes the proof. □

Proof of Theorem 2.3.4. Slightly abusing the notation we write $d(e)$ for a tuple $e = (z, f)$ by which we mean the distance $d(z, f)$. Combining Lemma 2.3.7 and 2.3.8 we obtain

$$\mathsf{med}(\widetilde{C}, \widetilde{\sigma}) \leq \sum_{c \in \widetilde{C}} \Bigg(\sum_{y \in P_c \setminus N_c} d(y, c) + \sum_{y \in N_c} \alpha^2 (d(y, d_y) + d(d_y, x_y)) + \alpha d(x_y, c) \Bigg) \tag{2.3}$$

$$= \sum_{e \in T0} d(e) + \alpha^2 \sum_{e \in T2} d(e) + \alpha \sum_{e \in T1} d(e) \tag{2.4}$$

$$\leq (\alpha^2 + \alpha) \mathsf{med}(C, a). \tag{2.5}$$

By Lemma 2.3.7 we know that a tuple only appears at most once as any of the three tuple types. We replace (2.3) by summing up the cost of all tuples in Ti for $i = 0, 1, 2$ with the respective factor for each type and obtain (2.4).

Finally by Observation 2.3.6 the cost $d(e)$ for $e \in T0 \cup T1 \cup T2$ occurs as a term in the original solution and $T0 \cap T1 \cap T2 = \emptyset$ by Lemma 2.3.8, which proves (2.5). □

In Section 2.3.1 we reduce generalized k-median with weak lower bounds to generalized k-median with center cost and obtain a $(6.5 + \epsilon)$ or $O(1)$-approximation for k-median or k-means with weak lower bounds, respectively. We combine this with Theorem 2.3.4 to get a solution with 2-weak lower bounds whose cost is a constant factor away from the problem with weak lower bounds. Since weak lower bounds are a relaxation of 2-weak lower bounds, we get:

Corollary 2.3.9. *Let $\mathcal{O}$ be an optimal solution to k-median/k-means with 2-weak lower bounds and $\epsilon > 0$ be a constant. We can compute a solution (C, σ) in polynomial time for*

1. *k-median with 2-weak lower bounds with* $\mathsf{med}(C, \sigma) \leq (13 + \epsilon)\mathsf{med}(\mathcal{O})$
2. *k-means with 2-weak lower bounds with* $\mathsf{mean}(C, \sigma) \leq O(1)\mathsf{mean}(\mathcal{O})$.

2.3.3 Reducing the Number of Assignments per Point to (1+ϵ)

So it is possible to reduce the number of assignments per point to two at a constant factor increase in the approximation factor. We can go even further and allow points to be fractionally assigned to centers which poses the question if it is possible to bound the assigned amount by a number smaller than two. Indeed we can prove for every $\epsilon \in (0, 1)$ that we can modify a solution to generalized k-median with weak lower bounds such that every point is assigned by an amount of at most $1 + \epsilon$ and the cost increases by a factor of $\mathcal{O}(\frac{1}{\epsilon}\alpha^2)$. Note that even if we allow fractional assignments of points to centers, the centers remain either open or closed, which differentiates our result from a truly fractional solution, where it is also allowed to open centers fractionally. Furthermore, the new assignment assigns every point to at most two centers. It is assigned by an amount of one to one center and potentially by an additional amount of ϵ to a second center. We say that such solutions satisfy $(1 + \epsilon)$-weak lower bounds.

Since we consider fractional assignments we modify our notation and denote with $\widetilde{\sigma}_x^c \in [0, 1]$ the amount by which $x \in \mathcal{P}$ is assigned to $c \in \widetilde{C}$, where $\widetilde{C}$ is the set of centers. Let $\widetilde{\sigma}_x = \sum_{c \in \widetilde{C}} \widetilde{\sigma}_x^c$ be the amount by which $x \in \mathcal{P}$ is assigned to $\widetilde{C}$. The assignment $\widetilde{\sigma}$ is feasible if $\widetilde{\sigma}_x \geq 1$ for all $x \in \mathcal{P}$ and $\sum_{x \in \mathcal{P}} \widetilde{\sigma}_x^c \geq B(c)$ for all $c \in \widetilde{C}$, and its cost is

$$\mathsf{med}(\widetilde{C}, \widetilde{\sigma}) = \sum_{c \in \widetilde{C}} \sum_{x \in \mathcal{P}} \widetilde{\sigma}_x^c d(x, c).$$

The proof of the following theorem is similar to the proof of Theorem 2.3.4 but to satisfy lower bounds we can only assign an amount of ϵ from points which are already assigned once. Therefore we consider suitable sets with $\lceil \frac{1}{\epsilon} \rceil$ points, which leads to the increase of $\mathcal{O}(\frac{1}{\epsilon})$ in the approximation factor.

Theorem 2.3.10. *Given $0 < \epsilon < 1$ and a solution (C, σ) to generalized k-median with weak lower bounds, we can compute a solution $(\widetilde{C}, \widetilde{\sigma})$ to generalized k-median with $(1 + \epsilon)$-weak lower bounds in polynomial time such that* $\mathsf{med}(\widetilde{C}, \widetilde{\sigma}) \leq (\lceil \frac{1}{\epsilon} \rceil \alpha(\alpha + 1) + 1)\mathsf{med}(C, \sigma)$.

Reassignment process. In the beginning we set $\widetilde{C} = C$. For $q \in \mathcal{P}$ let $\widetilde{\sigma}_q^c = 1$ if $c \in \sigma(q)$ and otherwise let $\widetilde{\sigma}_q^c = 0$. We modify both $\widetilde{C}$ and $\widetilde{\sigma}$ until we obtain a valid solution to generalized k-median with $(1 + \epsilon)$-weak lower bounds. During the process, the centers in $\widetilde{C}$ are called *currently open*, and when a center is deleted from $\widetilde{C}$, we say it is *closed*. The centers are processed in an arbitrary but fixed order, i.e., we assume that $\widetilde{C} = C = \{c_1, \ldots, c_{k'}\}$ for some $k' \leq k$ and process them in order $c_1, \ldots, c_{k'}$. We say that c_i is *smaller* than c_j if $i < j$.

Before we start explaining the reassignment we observe that the following properties hold for $(\widetilde{C}, \widetilde{\sigma})$ in the beginning.

1. for all $q \in \mathcal{P}$ we have either $\widetilde{\sigma}_q \in \mathbb{N}$ or $\widetilde{\sigma}_q = 1 + \epsilon$.

2. if $\widetilde{\sigma}_q = 1 + \epsilon$ then q is assigned to one center by an amount of one and to a second center by an amount of ϵ.

3. if $\widetilde{\sigma}_q \in \mathbb{N}$ then $\widetilde{\sigma}_q^c \in \{0, 1\}$ for all $c \in \widetilde{C}$.

We ensure that these properties also hold during the whole reassignment process.

Let $c = c_i$ be the currently processed center. By P_c we denote the set of points assigned to c by a positive amount under $\widetilde{\sigma}$. We divide P_c into the four sets $P_c^1 = \{q \in P_c \mid \widetilde{\sigma}_q = \widetilde{\sigma}_q^c = 1\}$, $P_c^\epsilon = \{q \in P_c \mid \widetilde{\sigma}_q = 1 + \epsilon, \widetilde{\sigma}_q^c = \epsilon\}$, $Q_c^\epsilon = \{q \in P_c \mid \widetilde{\sigma}_q = 1 + \epsilon, \widetilde{\sigma}_q^c = 1\}$ and finally $P_c^2 = \{q \in P_c \mid \widetilde{\sigma}_q \geq 2, \widetilde{\sigma}_q^c = 1\}$. Thus we differentiate between points which are assigned exclusively to c, points which are assigned by an amount of ϵ to c and by an amount of one to an other center or vice versa and points which are assigned by an amount of one to c and by an amount of at least one to some other centers. Furthermore with $C(P_c^2)$ we denote all centers which are connected to at least one point in P_c^2 under $\widetilde{\sigma}$. Observe that indeed $P_c = P_c^1 \cup P_c^\epsilon \cup Q_c^\epsilon \cup P_c^2$ if the above properties hold at that time.

Notice that points in $P_c \backslash P_c^2$ are already assigned by an amount of at most $1 + \epsilon$, so we only care about points in P_c^2. If P_c^2 is empty, we are done and proceed with the next center in $\widetilde{C}$. Otherwise we need to empty P_c^2. Observe that points in P_c^2 are assigned to multiple centers, so if we delete the connection between one of these points and c, the point is still served by some other center. However, doing so violates the lower bound at c. So we have to replace this connection.

As long as P_c^2 is non-empty, we do the following. We pick a center $d = \min C(P_c^2) \backslash \{c\}$ and a point $x \in P_c^2$ connected to d. We want to assign points from P_d^1 by amount of ϵ to c to free x. For technical reasons, we restrict the choice of these points: We exclude all points from the subset $\overline{P_d^1} := \{q \in P_d^1 \mid |\sigma(q)| \geq 2 \text{ and } \sigma(q) \cap \{c_1, \ldots, c_{i-1}\} \cap \widetilde{C} \neq \emptyset\}$, i.e., all points which were assigned to at least 2 centers under the initial assignment σ, and where one of these at least 2 centers is still open *and* smaller than c.

We can only assign points from $P_d^1 \backslash \overline{P_d^1}$ to c if its cardinality is at least $\lceil \frac{1}{\epsilon} \rceil$. If this is the case we choose a set A of $\lceil \frac{1}{\epsilon} \rceil$ points from $P_d^1 \backslash \overline{P_d^1}$ and set $\widetilde{\sigma}_q^c = \epsilon$ for all $q \in A$. Furthermore we set $\widetilde{\sigma}_x^c = 0$. So x is no longer connected to c, but to satisfy the lower bound at c we replace x by a set of $\lceil \frac{1}{\epsilon} \rceil$ points which are now connected to c by an amount of ϵ (Figure 2.3). By this we guarantee that the lower bound at c is still satisfied.

If $|P_d^1 \backslash \overline{P_d^1}| < \lceil \frac{1}{\epsilon} \rceil$ our replacement plan does not work. Instead we close d and set $\widetilde{\sigma}_q^d = 0$ for all $q \in \mathcal{P}$. If we close d, points in $P_d^1 \cup Q_d^\epsilon$ will be assigned by an amount smaller than one, thus we do the following. All points in $P_d^1 \backslash \overline{P_d^1}$ are reassigned to c, i.e., $\widetilde{\sigma}_q^c = 1$ for $q \in P_d^1 \backslash \overline{P_d^1}$ (Figure 2.4). Since we assign all points in $P_d^1 \backslash \overline{P_d^1}$ to c, we could delete this many connections between points in P_c^2 and c. But for simplicity, if $P_d^1 \backslash \overline{P_d^1}$ is non-empty and $\widetilde{\sigma}_x \geq 3$, we only delete the connection between x and c. A point $q \in \overline{P_d^1}$ is reassigned to the smallest open center in $\sigma(q)$ by an amount of one. And finally every point in Q_d^ϵ is assigned by an amount of ϵ to some other center than d, so we add an additional amount of $1 - \epsilon$ to this assignment.

Observe that none of the above reassignments violates the claimed properties for $(\widetilde{C}, \widetilde{\sigma})$ above. The entire procedure is described in Algorithm 2.

Lemma 2.3.11. *Algorithm 2 computes a feasible solution $(\widetilde{C}, \widetilde{\sigma})$ to generalized k-median with $(1 + \epsilon)$-weak lower bounds. Furthermore the following properties hold during all steps of the algorithm.*

Algorithm 2: Reducing the number of assigned centers per point to $1+\epsilon$

define an ordering on the centers $c_1 < c_2 \ldots < c_{k'}$
set $\widetilde{C} := C$ and $\widetilde{\sigma}_q^c = 1$ if $c \in \sigma(q)$ otherwise set $\widetilde{\sigma}_q^c = 0$
for all $c \in C$
    $P_c = \{q \in \mathcal{P} \mid \widetilde{\sigma}_q^c > 0\}$
    $P_c^1 = \{q \in P_c \mid \widetilde{\sigma}_q = \widetilde{\sigma}_q^c = 1\}$
    $P_c^\epsilon = \{q \in P_c \mid \widetilde{\sigma}_q = 1+\epsilon,\, \widetilde{\sigma}_q^c = \epsilon\}$
    $Q_c^\epsilon = \{q \in P_c \mid \widetilde{\sigma}_q = 1+\epsilon,\, \widetilde{\sigma}_q^c = 1\}$
    $P_c^2 = \{q \in P_c \mid \widetilde{\sigma}_q \geq 2,\, \widetilde{\sigma}_q^c = 1\}$
for $i = 1$ **to** k' **do**
    while $P_{c_i}^2 \neq \emptyset$ **do**
        $d = \min C(P_{c_i}^2) \backslash \{c_i\}$
        $\overline{P_d^1} = P_d^1 \cap \{q \in \mathcal{P} \mid |\sigma(q)| \geq 2 \text{ and } \sigma(q) \cap \{c_1, \ldots, c_{i-1}\} \cap \widetilde{C} \neq \emptyset\}$
        if $|P_d^1 \backslash \overline{P_d^1}| < \frac{1}{\epsilon}$ **then**
            delete d from $\widetilde{C}$ and all connections to d in $\widetilde{\sigma}$
            for all $q \in \overline{P_d^1}$
                let $e = \min(\sigma(q) \cap \widetilde{C})$
                set $\widetilde{\sigma}_q^e = 1$
            for all $q \in Q_d^\epsilon$
                let $e \in \widetilde{C}$ such that $\widetilde{\sigma}_q^e = \epsilon$
                set $\widetilde{\sigma}_q^e = 1$
            if $P_d^1 \backslash \overline{P_d^1} \neq \emptyset$ **then**
                pick $x \in P_{c_i}^2$ connected to d
                if $\widetilde{\sigma}_x \geq 3$ set $\widetilde{\sigma}_x^{c_i} = 0$
                for all $q \in P_d^1 \backslash \overline{P_d^1}$
                    set $\widetilde{\sigma}_q^{c_i} = 1$
        else
            pick $x \in P_{c_i}^2$ connected to d and $A \subset P_d^1 \backslash \overline{P_d^1}$ of cardinality $\lceil \frac{1}{\epsilon} \rceil$
            set $\widetilde{\sigma}_x^{c_i} = 0$ and $\widetilde{\sigma}_y^{c_i} = \epsilon$ for all $y \in A$

1. *For any center $c \in C$, P_c does not change before c is processed or closed. Up to that point all points in P_c are assigned by an amount of 1 to c.*

2. *If a connection between $x \in \mathcal{P}$ and the currently processed center $c \in \widetilde{C}$ is deleted by the algorithm, we have from this time on $x \notin P_c^2$ until termination. Moreover P_c^2 remains empty after c is processed.*

3. *While the algorithm processes $c \in C$ we always have $c < \min C(P_c^2)\backslash\{c\}$. Moreover all currently open centers which are smaller than c remain open until termination.*

4. *If the algorithm establishes a new connection in Line 17, Line 25 or Line 28 it remains until termination.*

Proof. The process terminates: For every iteration of the while loop starting in Line 10, either a point is deleted from $P_{c_i}^2$ or there is at least one point $x \in P_{c_i}^2$ for which $\widetilde{\sigma}_x$ is reduced by one. Furthermore $\widetilde{\sigma}_x$ does never increase for any $x \in P_{c_i}^2$.

The final solution satisfies lower bounds: Every time we delete a connection between a point and a center it either happens because the center is closed or we replace this connection by assigning $\lceil \frac{1}{\epsilon} \rceil$ new points each by an amount of ϵ to it. So the lower bounds are satisfied at all open centers.

All points are assigned by an amount of at least 1: Assume that the algorithm deletes the connection between a point p and a center d. This either happens if p is assigned by a total amount of at least 2 at this time or d is closed by the algorithm. In the last case we ensure in Line 17, Line 20 or Line 25 that p is assigned by an amount of one to an other center after we close d.

All points are assigned by an amount of at most $1 + \epsilon$: For $c \in C$ we know by Property 2 that P_c^2 is empty after termination. Then $P_c = P_c^1 \cup P_c^\epsilon \cup Q_c^\epsilon$, so all points connected to c are assigned by a total amount of at most $1 + \epsilon$.

We conclude that the solution is feasible.

Property 1: Let $c \in C$. Assume the property is true up to a time t. In the next step connections may change for the center that is currently processed, for a smaller center which has been processed already or for a center which is currently connected to a point by an amount of ϵ. If c is not processed so far none of this applies to it, so the property also holds in the next step.

Property 2: Assume that after the connection between $x \in P_c$ and c is deleted by the algorithm, x is part of P_c^2. That would require that the algorithm assigns x to a center by an amount of one while it is already assigned to a second center by an amount of one, which does not happen. For the same reason P_c^2 remains empty after c is processed by the algorithm.

Property 3: Assume c is currently processed by the algorithm and $d = \min C(P_c^2)\backslash\{c\}$. We know that at this time P_d^2 is non-empty. Which is by Property 2 only possible if d is processed after c. Thus we have $c < d$. This also means that centers can only be closed by the algorithm if they are not processed so far.

Property 4: A connection established in Line 17 involves a center which is already processed by the algorithm. By Property 3 such centers remain open, thus the connection is not deleted until termination. In Line 25 and Line 28 the algorithm establishes a connection between the currently processed center c and some point p which is assigned by an amount of at most 1 at this time. If this connection is deleted at some later point in time, this would require that c is closed by the algorithm or $p \in P_c^2$. Both can not happen. □

We bound the cost of $(\widetilde{C}, \widetilde{\sigma})$ in a similar way we bounded the cost of the solution in Theorem 2.3.4. Let N_c denote the set of points which are newly assigned by ϵ respectively 1 to c while c is processed. This happens in Line 25 and Line 28 of the algorithm. We want to charge the cost of these new connections to the cost of the original solution.

For $y \in N_c$ let d_y be the respective center in Line 11 of Algorithm 2 and x_y the point in Line 22 respectively Line 27 contained in P_c^2 and connected to d_y. Using the α-relaxed triangle inequality, we obtain the following upper bound.

$$\begin{aligned} d(y,c) &\leq \alpha(d(y,x_y) + d(x_y,c)) \leq \alpha\Big(\alpha\Big(d(y,d_y) + d(d_y,x_y)\Big) + d(x_y,c)\Big) \\ &\leq \alpha^2\Big(d(y,d_y) + d(d_y,x_y)\Big) + \alpha d(x_y,c). \end{aligned} \tag{2.6}$$

We can apply (2.6) to all $c \in \widetilde{C}$ and all $y \in N_c$. This yields the following upper bound on the cost of the final solution $(\widetilde{C}, \widetilde{\sigma})$.

$$\begin{aligned} \mathsf{med}(\widetilde{C}, \widetilde{\sigma}) &= \sum_{c \in \widetilde{C}} \sum_{y \in \mathcal{P}} d(y,c)\widetilde{\sigma}_y^c \leq \sum_{c \in \widetilde{C}} \Big(\sum_{y \in P_c \setminus N_c} d(y,c) + \sum_{y \in N_c} d(y,c) \Big) \\ &\leq \sum_{c \in \widetilde{C}} \Big(\sum_{y \in P_c \setminus N_c} d(y,c) + \sum_{y \in N_c} \alpha^2(d(y,d_y) + d(d_y,x_y)) + \alpha d(x_y,c) \Big). \end{aligned} \tag{2.7}$$

Notice that in the first inequality we use the fact that $\widetilde{\sigma}_y^c \leq 1$. So we pay the the price of connecting y to c by an amount of 1 independent of whether $\widetilde{\sigma}_y^c$ is 1 or ϵ.

Expression (2.7) is what we want to pay for. Observe that all involved distances contribute to the original cost as well (we state this formally in Observation 2.3.12 below). So in principle, we can charge each summand to a term in the original cost. But what we need to do is to bound the number of times that each term in the original cost gets charged. To organize the counting, we count how many times a specific tuple of a point z and a center f occurs as $d(z,f)$ in (2.7). Since it is important at which position a tuple appears, we give names to the different occurrences. We say that that a tuple appears as a tuple of Type 0 if it appears as $d(y,c)$ in (2.7), as tuple of Type 1 if it appears as $d(x_y,c)$, and as tuple of Type 2 if it appears as $d(y,d_y)$ or $d(d_y,x_y)$. We distinct the latter type further by calling a tuple occurring as $d(y,d_y)$ a tuple of Type 2.1 and a tuple occurring as $d(x_y,d_y)$ a tuple of Type 2.2. We say that $(y,d_y),(d_y,x_y)$ and (x_y,c) *contribute* to the cost of (y,c), where by the *cost* of (y,c) we mean the upper bound on $d(y,c)$ in (2.6) which we want to pay for.

Observation 2.3.12. *If a tuple (z,f), $z \in \mathcal{P}, f \in C$, occurs as Type 0, 1 or 2, then $f \in \sigma(z)$, so in particular, $d(z,f)$ occurs as a term in the cost of the original solution.*

Proof. For a center c the set $P_c \setminus N_c$ contains points which are assigned to c by the initial assignment σ or assigned to c while c is not processed by the algorithm. Latter can only happen if a connection is reestablished in Line 17 which requires that the connection was already present in (C,σ). So Type 0 tuples satisfy the statement.

For Type 1 and 2 tuples, consider $y \in N_c$ for some center c and the respective tuples $(x_y,c),(y,d_y)$, (x_y,d_y). Notice that both y and x_y are connected to d_y before y is assigned to c. By Property 3 of Lemma 2.3.11 we have $c < d_y$. Thus we know by Property 1 of Lemma 2.3.11 that $d_y \in \sigma(y)$ and $d_y \in \sigma(x_y)$ which proves that Type 2 tuples satisfy the statement. Moreover it holds that $c \in \sigma(x_y)$ since there is a time where $x_y \in P_c^2$, which can only happen if the connection between x_y and c is already part of (C,σ). Thus, Type 1 tuples satisfy the statement. □

As indicated above, a tuple (z, f) can contribute to the cost of multiple tuples. Notice that a tuple occurs at most once as a tuple of Type 0 in (2.7). To bound the cost of $(\widetilde{C}, \widetilde{\sigma})$ we bound the number of times a tuple appears as Type 1 or Type 2 tuple in (2.7).

Remember that we used a similar statement in the proof of Theorem 2.3.4, where we proved that every tuple can appear at most once as each type. However here we can only bound the appearance by $\lceil \frac{1}{\epsilon} \rceil$ for Type 1 and Type 2 tuples due to Line 25 and Line 28 where we assign up to $\lceil \frac{1}{\epsilon} \rceil$ points from P_d^1 to c. Notice that even if we assign each of these points initially by an amount of ϵ to c as it is done in Line 28, that amount can be increased to 1 at some later time in Line 20. The proof is similar to that of Lemma 2.3.7 but we carry out the arguments again for sake of completeness.

Lemma 2.3.13. *For all $z \in \mathcal{P}, f \in C$, the tuple (z, f) appears in (2.7) at most $\lceil \frac{1}{\epsilon} \rceil$ times as tuple of Type 1 and at most $\lceil \frac{1}{\epsilon} \rceil$ times as tuple of Type 2.*

Proof. In the following, the tuple whose cost the tuple (z, f) contributes to will always be named (y, c), and we denote the time at which y is newly assigned to c by t.

Type 1: Assume (z, f) contributes to the cost of (y, c) as a Tuple of Type 1. Then $f = c$. At t we assign up to $\lceil \frac{1}{\epsilon} \rceil$ points to c. So (z, f) contributes to the cost of at most $\lceil \frac{1}{\epsilon} \rceil$ connections established by the algorithm at t as tuple of Type 1. Notice that at the time step before t we must have $z \in P_c^2$ and afterwards, z is never again contained in P_c^2 by Property 2 of Lemma 2.3.11. Thus the tuple (z, c) can not be responsible for any assignment to c after t, i.e., $(z, c) = (z, f)$ does not contribute to any further cost as a tuple of Type 1.

Type 2.1: Assume that (z, f) contributes to the cost of (y, c) as a Tuple of Type 2.1. Then $z = y$. At the time step before t, we have $y \in P_f^1$, $f \in C(P_c^2)$. By Property 4 of Lemma 2.3.11, newly established connections stay, so after time t, it always holds that $y \in P_c$. So even if y is in P_f at a later time, it can not be in P_f^1 since it is also connected to c. So $(y, f) = (z, f)$ does not contribute to any further cost as tuple of Type 2.1. Furthermore, observe that the algorithm never adds a connection to a point which is assigned more than once. So we know that y is always assigned by an amount of at most $1 + \epsilon$ after t which means that (y, f) does not contribute as tuple of Type 2.2 to the cost of any connection established by the algorithm after t either.

Type 2.2: Finally we consider the case where (z, f) contributes to the cost of (y, c) as a tuple of Type 2.2. At time t, the algorithm processes c. By the way the algorithm chooses f and z, we know that $z \in P_c^2$ (at the beginning of the process, i.e., before t) and $f = \min C(P_c^2) \backslash \{c\}$. After t, Property 2 of Lemma 2.3.11 implies $z \notin P_c^2$, which means that as a tuple of Type 2.2, it can not contribute to the cost of any tuple containing c after t. However it contributes as tuple of Type 2.2 to the cost of up to $\lceil \frac{1}{\epsilon} \rceil - 1$ additional connections at time t (see Line 25 and Line 28). Assume instead that it contributes (as Type 2.2) to the cost of a tuple (y', c') for a center $c' \neq c$, and some point $y' \in \mathcal{P}$. This is supposed to happen after t, so y' is newly assigned to c' at some time $t' > t$. The step before t' we have $z \in P_{c'}^2$. Thus before c' is processed, we must always have $z \in P_{c'}^2$ by Property 1 of Lemma 2.3.11. So in particular, at time $t < t'$ we have $c' \in C(P_c^2) \backslash \{c\}$. Moreover we know that at some time while c' is processed by the algorithm we have $f = \min C(P_{c'}^2) \backslash \{c'\}$. Using Property 3 of Lemma 2.3.11 we conclude that $c' < f$. Which is a contradiction since the algorithm chose f and not c' at time t, i.e., $f = \min C(P_c^2) \backslash \{c\}$ must hold. Thus, (z, f) can not contribute to the cost of (y', c') as a tuple of Type 2.2.

It is left to show that (z, f) can not contribute to the cost of any (y', c') as a tuple of Type 2.1 at some time $t' > t$. For a contribution as Type 2.1, we would have $z = y'$ and

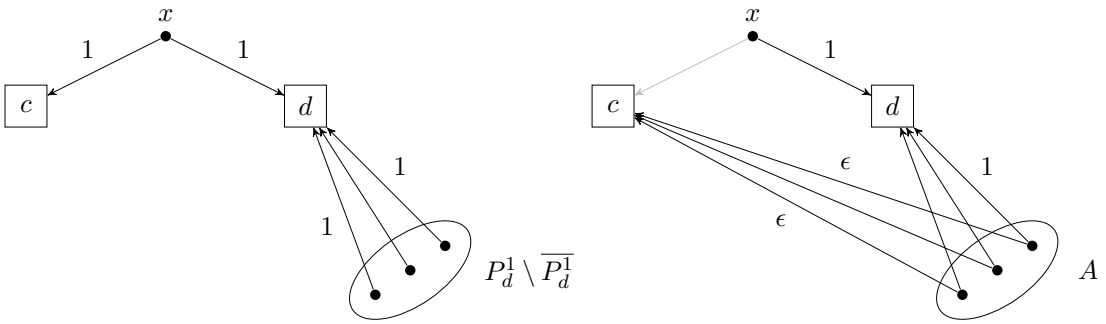

Figure 2.3: Shows case $|P_d^1 \setminus \overline{P_d^1}| > \lceil \frac{1}{\epsilon} \rceil$. Pick a set $A \subset P_d^1 \setminus \overline{P_d^1}$ of cardinality $\lceil \frac{1}{\epsilon} \rceil$ and assign an amount of ϵ from points in A to c. Here $A = P_d^1 \setminus \overline{P_d^1}$.

$y' \in P_f^1$. We show that in this case y' is even contained in $\overline{P_f^1}$. Remember that at time t we have $y' = z \in P_c^2$ and that this only happens if $|\sigma(y')| \geq 2$. Moreover c is sill open by Property 3 of Lemma 2.3.11 and is smaller than c'. Thus $c \in \sigma(y') \cap \{e \mid e < c'\} \cap \widetilde{C}$, which proves $y' \in \overline{P_f^1}$. Therefore the algorithm does not assign y' to c' (see Line 17) and (z, f) does not contribute as tuple of Type 2.1 to the cost of any connection established by the algorithm after t. □

For the final counting, we define $T0$, $T1$ and $T2$ as the sets of all tuples of Type 0, 1 and 2, respectively.

Proof of Theorem 2.3.4. Slightly abusing the notation we write $d(e)$ for a tuple $e = (z, f)$ by which we mean the distance $d(z, f)$. We obtain

$$\mathsf{med}(\widetilde{C}, \widetilde{\sigma}) \leq \sum_{c \in \widetilde{C}} \Big(\sum_{y \in P_c \setminus N_c} d(y, c) + \sum_{y \in N_c} \alpha^2 (d(y, d_y) + d(d_y, x_y)) + \alpha d(x_y, c) \Big) \tag{2.7}$$

$$= \sum_{e \in T0} d(e) + \alpha^2 \left\lceil \frac{1}{\epsilon} \right\rceil \sum_{e \in T2} d(e) + \alpha \left\lceil \frac{1}{\epsilon} \right\rceil \sum_{e \in T1} d(e) \tag{2.8}$$

$$\leq \Big(\left\lceil \frac{1}{\epsilon} \right\rceil \alpha(\alpha + 1) + 1 \Big) \mathsf{med}(C, a). \tag{2.9}$$

Here we replace (2.7) by summing up the cost of all tuples in Ti for $i = 0, 1, 2$ with the respective factor times the maximal number of appearances for each type. Thus by Lemma 2.3.13 we obtain a total factor of 1 for Type 0, $\alpha^2 \lceil \frac{1}{\epsilon} \rceil$ for Type 1 and $\alpha \lceil \frac{1}{\epsilon} \rceil$ for Type 2 (see (2.8)).

Finally by Observation 2.3.12 the cost $d(e)$ for $e \in T0 \cup T1 \cup T2$ occurs as a term in the original solution which proves (2.9). □

Note that we also prove that we can find a fractional assignment of a special structure. The assignment $\widetilde{\sigma}$ assigns every point to at most two centers. It is assigned by an amount on one to one center and eventually by an additional amount of ϵ to a second center.

Combining the results in Section 2.3.1 with Theorem 2.3.10 we obtain:

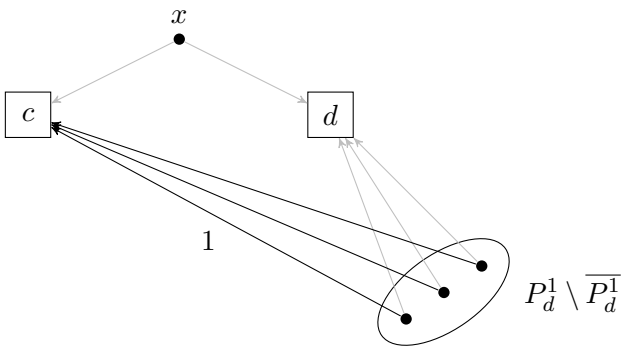

Figure 2.4: Shows case $|P_d^1 \setminus \overline{P_d^1}| < \lceil \frac{1}{\epsilon} \rceil$. Center d is closed and points from $P_d^1 \setminus \overline{P_d^1}$ are assigned to c.

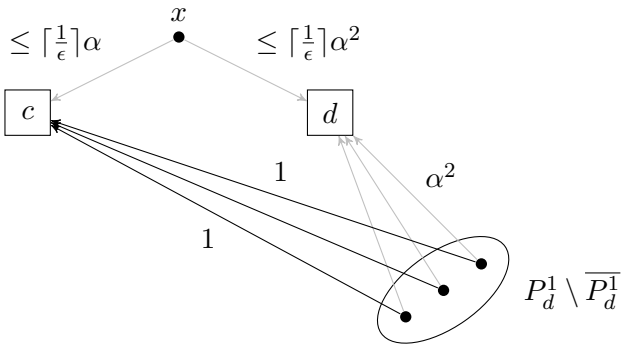

Figure 2.5: Showing the case where d is closed. To bound the distance from points in $P_d^1 \setminus \overline{P_d^1}$ to c the respective distances appear with a factor of $\lceil \frac{1}{\epsilon} \rceil \alpha$, $\lceil \frac{1}{\epsilon} \rceil \alpha^2$ or α^2.

Corollary 2.3.14. *Let $\mathcal{O}$ be an optimal solution to k-median/k-means with $(1+\epsilon')$-weak lower bounds and $\epsilon > 0$ be a constant. We can compute a solution (C, σ) in polynomial time for*

1. *k-median with $(1+\epsilon')$-weak lower bounds with $\mathsf{med}(C,\sigma) \leq ((13+\epsilon)\lceil \frac{1}{\epsilon'} \rceil + 6.5 + \epsilon)\mathsf{med}(\mathcal{O})$*
2. *k-means with $(1+\epsilon')$-weak lower bounds with $\mathsf{mean}(C,\sigma) \leq O(\frac{1}{\epsilon'})\mathsf{mean}(\mathcal{O})$.*

2.4 A Bi-Criteria Algorithm to Generalized k-Median with Lower Bounds

So far we presented an algorithm that computes a set of at most k centers $C \subset \mathcal{X}$ and an assignment $\sigma \colon \mathcal{P} \to 2^C$ such that the lower bound is satisfied at all centers and every point is assigned at least once and at most twice. Instead allowing a point to be assigned multiple times to satisfy the lower bounds, we can instead allow to violate the lower bound by some fixed factor. Such solutions are called bi-criteria approximations and are defined as follows.

Definition 2.4.1. *A (β, δ)-bi-criteria solution for generalized k-median with lower bounds consists of at most k centers $C \subset \mathcal{X}$ and an assignment $\sigma \colon \mathcal{P} \to C$ such that at least*

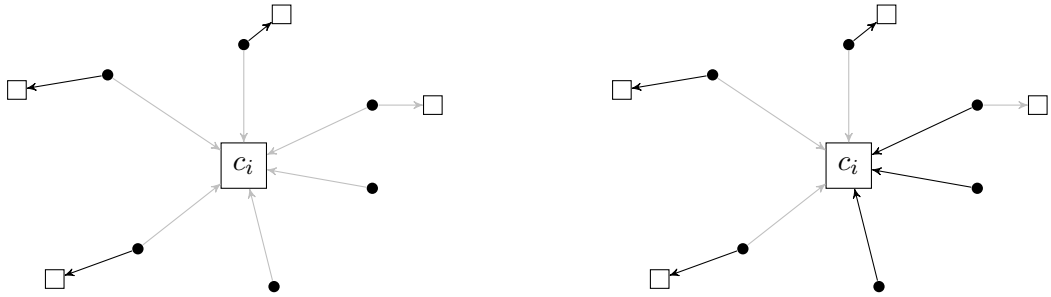

Figure 2.6: Shows the case where A_i contains at least $\lceil \beta B(c_i) \rceil$ unassigned points. The three points on the left are already assigned to other centers and the three points on the right are newly assigned to c_i. The gray connections come from σ.

$\beta B(c)$ points are assigned to $c \in C$ by σ and $\mathsf{med}(C, \sigma) \leq \delta \mathsf{med}(\mathcal{O})$. Here $\mathcal{O}$ denotes an optimal solution to generalized k-median with lower bounds.

Guha, Meyerson and Munagala [49] and Karger, Minkoff [55] presented a bi-criteria algorithm for facility location with lower bounds which can easily be transformed into a bi-criteria algorithm for k-median with lower bounds. However their approach only works for distances which are given be a metric, especially it does not yield a constant factor bi-criteria approximation for k-means with lower bounds. In this section we use our results on 2-weak lower bounds to compute a bi-criteria approximation for generalized k-means with lower bounds.

Theorem 2.4.2. *Given a γ-approximate solution (C, σ) to generalized k-median with 2-weak lower bounds and a fixed $\beta \in [0.5, 1)$ we can compute a $(\beta, \gamma \max\{\frac{\alpha\beta}{1-\beta} + 1, \frac{\alpha^2\beta}{1-\beta}\})$-bi-criteria solution to generalized k-median with lower bounds in polynomial time. In particular, there exists a polynomial time $(\frac{1}{2}, O(1))$-bi-criteria approximation algorithm for k-means with lower bounds.*

Proof. Given a $\beta \geq \frac{1}{2}$ and a γ-approximate solution to generalized k-median with 2-weak lower bounds (C, σ), we can compute a $(\beta, \gamma \max\{\frac{\alpha\beta}{1-\beta} + 1, \frac{\alpha^2\beta}{1-\beta}\})$-bi-criteria solution in the following way. Let $C = \{c_1, \ldots, c_{k'}\}$ for some $k' \leq k$. We process the centers in order $c_1, \ldots, c_{k'}$ and decide if they are open or closed. We say that c_i is *smaller* than c_j if $i < j$. If we decide that a center c is open we directly assign at least $\lceil \beta B(c) \rceil$ points to c. In the beginning all points are unassigned.

Consider center c_i. Let A_i be the set of all points assigned to c_i under σ. We know that $|A_i| \geq B(c_i)$. If at least $\lceil \beta B(c_i) \rceil$ points in A_i are not assigned so far, c_i remains open and all currently unassigned points from A_i are assigned to c_i (Figure 2.6). If fewer than $\lceil \beta B(c_i) \rceil$ points from A_i are unassigned, the center is closed.

Let C' denote the centers from $\{c_1, \ldots, c_{i-1}\}$ which are open and B_i the set of unassigned points from A_i which are not connected to any center larger than c_i under σ. To guarantee that all points are assigned at the end, we have to take care of points in B_i. By assumption there are at most $\lfloor \beta B(c_i) \rfloor$ such points. We simply assign any point $p \in B_i$ to the nearest center $\arg\min_{c \in C'} d(c, p)$ in C'. The whole procedure is described in Algorithm 3.

To upper bound the assignment cost in the case c_i is closed by the algorithm we consider a second assignment τ, which may be fractional. We define for $p \in B_i$ and

$c \in C'$ a value $\tau_p^c \in [0,1]$ which indicates the amount by which p is assigned to c. We claim that we can find a fractional assignment such that for every $q \in B_i$ and $f \in C'$ the following holds

1. point q is assigned by an amount of one, i.e.,

$$\sum_{c \in C'} \tau_q^c = 1$$

2. and at most $\frac{\beta}{1-\beta}|\{p \in A_i \mid f \in \sigma(p)\}|$ amount is assigned to f, i.e.,

$$\sum_{p \in B_i} \tau_p^f \leq \frac{\beta}{1-\beta}|\{p \in A_i \mid f \in \sigma(p)\}|$$

Such an assignment can be found since

$$\begin{aligned}\frac{\beta}{1-\beta}\sum_{c \in C'}|\{p \in A_i \mid c \in \sigma(p)\}| =& \frac{\beta}{1-\beta}|\{p \in A_i \mid \sigma(p) \cap C' \neq \emptyset\}| \\ \geq& \frac{\beta(1-\beta)}{1-\beta}B(c_i) \geq |B_i|.\end{aligned}$$

To see the first inequality we observe the following. If a point $p \in A_i$ is connected to an open center $c \in C'$ under σ, it is already assigned to c by the algorithm. So the set of points from A_i which are already assigned to some center equals $\{p \in A_i \mid \sigma(p) \cap C' \neq \emptyset\}$. We know that $|A_i| \geq B(c_i)$ and that at most $\lfloor \beta B(c_i) \rfloor$ points from A_i are unassigned. Thus we have $|\{p \in A_i \mid \sigma(p) \cap C' \neq \emptyset\}| \geq (1-\beta)B(c_i)$.

Let τ be an assignment satisfying the above properties. We obtain the following upper bound to the cost of τ.

$$\begin{aligned}\sum_{c \in C'}\sum_{p \in B_i} \tau_p^c \, d(p,c) \leq& \sum_{c \in C'}\Bigg(\sum_{x \in A_i \colon c \in \sigma(x)} \frac{\beta}{1-\beta}\Big(\alpha^2 d(c_i,x) + \alpha d(x,c)\Big) + \sum_{p \in B_i} \tau_p^c \alpha^2 d(p,c_i)\Bigg) \\ \leq& \frac{\alpha\beta}{1-\beta}\sum_{c \in C'}\sum_{x \in A_i \colon c \in \sigma(x)} d(x,c) + \frac{\alpha^2\beta}{1-\beta}\sum_{x \in A_i} d(x,c_i).\end{aligned}$$

For the first inequality we used the above bound on $\sum_{p \in B_i} \tau_p^c$. We can charge every point in $\{x \in A_i \mid c \in \sigma(x)\}$ up to an amount of $\frac{\beta}{1-\beta}$ for the assignment cost of B_i to c. Assume such a point x gets charged by an amount of $\delta \leq \tau_p^c$ for the distance $d(p,c)$. We obtain the following upper bound on the cost

$$\delta d(p,c) \leq \delta(\alpha^2 d(p,c_i) + \alpha^2 d(c_i,x) + \alpha d(x,c)).$$

Thus in total the distance $d(p,c_i)$ appears with a factor of $\tau_p^c\alpha^2$, distance $d(c_i,x)$ with factor $\frac{\beta}{1-\beta}\alpha^2$ and $d(x,c)$ with factor $\frac{\beta}{1-\beta}\alpha$ in the upper bound on the assignment cost of B_i to c.

The second inequality follows immediately from $\frac{\beta}{1-\beta} \leq 1$, $\sum_{c \in C'} \tau_p^c = 1$ and $B_i \cap \{x \in A_i \mid \sigma(x) \cap C' \neq \emptyset\} = \emptyset$.

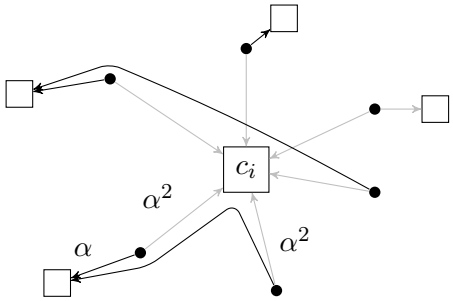

Figure 2.7: Showing assignment τ in the case where c_i is closed. The two points from B_i are distributed to centers in C'. The gray connections come from σ. α and α^2 are the factors with which the respective distances appear in the upper bound of the new connection.

Assigning every point in B_i to its nearest center can only be cheaper than distributing B_i to centers in C' via τ. We obtain

$$\begin{aligned}\sum_{p\in B_i}\min_{c\in C'} d(p,c) &\leq \sum_{c\in C'}\sum_{p\in B_i}\tau_p^c\, d(p,c)\\ &\leq \frac{\alpha\beta}{1-\beta}\sum_{c\in C'}\sum_{x\in A_i:\, c\in\sigma(x)} d(x,c) + \frac{\alpha^2\beta}{1-\beta}\sum_{x\in A_i} d(x,c_i). \qquad (2.10)\end{aligned}$$

Let (C',σ') be the final solution computed by the algorithm.

$$\begin{aligned}\mathsf{med}(C',\sigma') &= \sum_{c\in C'}\sum_{\substack{x\in\mathcal{P}:\\ \sigma'(x)=c}} d(x,c)\\ &\leq \sum_{c\in C'}\sum_{\substack{x\in\mathcal{P}:\\ c\in\sigma(x)}} d(x,c) + \frac{\alpha\beta}{1-\beta}\sum_{c\in C'}\sum_{\substack{x\in\mathcal{P}:\\ c\in\sigma(x)}} d(x,c) + \frac{\alpha^2\beta}{1-\beta}\sum_{c\in C\setminus C'}\sum_{\substack{x\in\mathcal{P}:\\ c\in\sigma(x)}} d(x,c)\\ &\leq \max\{\frac{\alpha\beta}{1-\beta}+1, \frac{\alpha^2\beta}{1-\beta}\}\sum_{c\in C}\sum_{\substack{x\in P:\\ c\in\sigma(x)}} d(x,c)\\ &= \max\{\frac{\alpha\beta}{1-\beta}+1, \frac{\alpha^2\beta}{1-\beta}\}\mathsf{med}(C,\sigma).\end{aligned}$$

To see the first inequality we use the upper bound in (2.10). Let $x\in\mathcal{P}$ and $c\in\sigma(x)$. If c is closed in the final solution the distance $d(x,c)$ is only charged with a factor of $\frac{\alpha^2\beta}{1-\beta}$ in (2.10) for closing c. If c is open in the final solution the distance $d(x,c)$ is charged with factor one if $\sigma'(x)=c$ and can also be charged with a factor of $\frac{\alpha\beta}{1-\beta}$ in (2.10) for closing a center $d\in\sigma(x)$. This can happen at most once since $|\sigma(x)|\leq 2$. This proves the second inequality.

Since generalized k-median with 2-weak lower bounds is a relaxation of generalized k-median with lower bounds we obtain

$$\mathsf{cost}(C',\sigma') \leq \gamma\max\{\frac{\alpha\beta}{1-\beta}+1, \frac{\alpha^2\beta}{1-\beta}\}\mathsf{cost}(\mathcal{O}).$$

Algorithm 3: A $(\beta, \gamma \max\{\frac{\alpha\beta}{1-\beta} + 1, \frac{\alpha^2\beta}{1-\beta}\})$-bi-criteria approximation algorithm to generalized k-median with lower bounds

Input : γ-approximate solution (C, σ) to generalized k-median with 2-weak lower bounds, $C = \{c_1, \ldots, c_{k'}\}$

Output: Bi-criteria solution (C', σ') to generalized k-median with lower bounds.

```
1  set C' = ∅, σ'(x) = ⊥ for all x ∈ P
2  N = P
3  for i = 1 to k' do
4      A_i = {x ∈ P | c_i ∈ σ(x)}
5      B_i = {x ∈ A_i | σ(x) ⊂ {c_1 ..., c_i}} ∩ N
6      if A_i ∩ N ≥ βB(c_i) then
7          set σ'(x) = c_i for all x ∈ A_i ∩ N
8          N = N\A_i
9          C' = C' ∪ {c_i}
10     else
11         set σ'(x) = arg min_{c∈C'} d(x, c) for all x ∈ B_i
```

This yields the $(\beta, \gamma \max\{\frac{\alpha\beta}{1-\beta} + 1, \frac{\alpha^2\beta}{1-\beta}\})$-bi-criteria approximation to generalized k-median with lower bounds. For k-means with lower bounds we apply Corollary 2.3.9 to see that there exists an $(\frac{1}{2}, O(1))$-bi-criteria algorithm with polynomial running time. □

Chapter 3

Hierarchical Clustering

The first part of this chapter contains results from the paper *The Price of Hierarchical Clustering* [14] by Anna Arutyunova and Heiko Röglin published in the proceedings of the *European Symposium on Algorithms (ESA), 2022*. A full version of this paper is available at *arXiv* [15].

The second part of this chapter contains results from the paper *Upper and Lower Bounds for Complete Linkage in General Metric Spaces* [13] by Anna Arutyunova, Anna Großwendt, Heiko Röglin, Melanie Schmidt and Julian Wargalla published in the proceedings of the *International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), 2021*. This paper is currently under consideration at the journal *Machine Learning*. This chapter extends [13] by an improved upper bound for complete linkage for the diameter, which was improved from $O(k^2)$ to $O(k^{1.59})$, and the notion of the average approximation factor.

In this chapter we consider the problem of computing hierarchical clusterings with respect to several objective functions. Given a set of n points $\mathcal{P}$ in a metric space $\mathcal{X}$ with metric d, the task is to compute a k-clustering for every $1 \leq k \leq n$ such that the clusterings are *hierarchicallly compatible*. Informally speaking two clusterings are hierarchically compatible if one is a refinement of the other. We say that a hierarchical clustering achieves an approximation factor α if every k-clustering is an α-approximation to the optimal k-clustering.

Remember that a k-clustering $\mathscr{C} = (\mathcal{C}_1, \dots, \mathcal{C}_k)$ is a partition of the set $\mathcal{P}$ into k clusters. The radius of the clustering is given by the maximal radius of one of its clusters. We consider two variants of the radius: the discrete radius, where the center of a cluster must be contained in the cluster itself, and the non-discrete radius, where the center can be any point in $\mathcal{X}$. More formally for a cluster $\mathcal{C} \subset \mathcal{P}$ the discrete radius is defined as

$$\mathsf{drad}(\mathcal{C}) = \min_{c \in \mathcal{C}} \max_{p \in \mathcal{C}} d(p, c)$$

and the discrete radius of the k-clustering $\mathscr{C}$ is given by $\mathsf{drad}(\mathscr{C}) = \max_{\mathcal{C} \in \mathscr{C}} \mathsf{drad}(\mathcal{C})$. The non-discrete radius of a cluster $\mathcal{C} \subset \mathcal{P}$ is defined as

$$\mathsf{rad}(\mathcal{C}) = \min_{c \in \mathcal{X}} \max_{p \in \mathcal{C}} d(p, c)$$

and the non-discrete radius of the k-clustering $\mathscr{C}$ is given by $\mathsf{rad}(\mathscr{C}) = \max_{\mathcal{C} \in \mathscr{C}} \mathsf{rad}(\mathcal{C})$. Furthermore the diameter of a cluster $\mathcal{C} \subset \mathcal{P}$ is defined as the maximum distance between two point in the cluster

$$\mathsf{diam}(\mathcal{C}) = \max_{p,q \in \mathcal{C}} d(p, q)$$

Figure 3.1: Here we see the optimal clusterings of size three and size two with respect to the diameter. These two clusterings are not hierarchically compatible.

and the diameter of the k-clustering $\mathscr{C}$ is given by $\mathsf{diam}(\mathscr{C}) = \max_{\mathcal{C}\in\mathscr{C}} \mathsf{diam}(\mathcal{C})$.

Now a hierarchical clustering and its approximation factor are defined as follows.

Definition 3.0.1. *Given an instance $(\mathcal{X}, \mathcal{P}, d)$, let $n = |\mathcal{P}|$. We call two clusterings $\mathcal{C}$ and $\mathcal{C}'$ of $\mathcal{P}$ with $|\mathcal{C}| \geq |\mathcal{C}'|$ hierarchically compatible if for all $C \in \mathcal{C}$ there exists $C' \in \mathcal{C}'$ with $C \subset C'$. A* hierarchical clustering *of $\mathcal{P}$ is a sequence of clusterings $\mathscr{H} = (\mathcal{H}_n, \ldots, \mathcal{H}_1)$, such that*

1. *$\mathcal{H}_i$ is an i-clustering of $\mathcal{P}$*
2. *for $1 < i \leq n$ the two clusterings $\mathcal{H}_{i-1}$ and $\mathcal{H}_i$ are hierarchically compatible.*

For $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}, \mathsf{drad}\}$ let $\mathcal{O}_i$ denote the optimal i-clustering with respect to cost. We say that $\mathscr{H}$ is an α-approximation with respect to cost if for all $i = 1, \ldots, n$ we have

$$\mathsf{cost}(\mathcal{H}_i) \leq \alpha \cdot \mathsf{cost}(\mathcal{O}_i).$$

Remember the example from the introduction shown in Figure 3.1 where we see that optimal clusterings do not have to be hierarchically compatible and therefore $\alpha = 1$ is usually not possible.

Given a clustering objective we can now ask the following two questions: What is the best polynomial time approximation algorithm unless $\mathsf{P} = \mathsf{NP}$? If given unlimited computation time, what is the best approximation factor that we can achieve? We consider the three objectives discrete radius, non-discrete radius and diameter. Regarding the first question, the best known algorithms are by Dasgupta and Long [39] and Charikar et al. [27] and both achieve an 8-approximation for all three objectives. Mondal [66] claims that there exists a 6-approximation for the discrete radius. The algorithm presented in [66] is a modification of the algortihm in [39]. However we present an instance in Section 3.1.3 where the algorithm in [66] does not achieve this approximation factor.

As we see in Figure 3.1 the answer to the second question has to be greater or equal to $\frac{3}{2}$ when considering the diameter. In the first part of this chapter we compute the best possible approximation factor for the objectives discrete radius, non-discrete radius and diameter.

Even though there exist polynomial time constant factor approximations for most of the popular clustering objectives such as the algorithms in [39, 27], agglomerative clustering methods are much more popular in practice. The complete linkage algorithm is such an agglomerative clustering method which starts with all points in separate clusters and in every step merges the two clusters whose cost results in the smallest increase of the considered objective function. Originally this algorithm seeks to minimize the diameter of emerging clusters in every step, but we also consider the variant where the algorithm minimizes the radius and call it also complete linkage for convenience. We analyze the performance guarantee of complete linkage in general metric spaces. So far it is only known that complete linkage may produce a hierarchical clustering which is not

better than a $\log_2(k)$-approximation [39] for the diameter. We provide upper bounds for complete linkage with respect to radius and diameter and also show examples where complete linkage computes a hierarchical clustering which is by factor $\Omega(k)$ worse than the optimal clustering for both radius and diameter.

We know that complete linkage performs reasonably well in the Euclidean space. Ackerman et al. [1] show an approximation guarantee of $O(\log(k))$ for all three objectives assuming the dimension of the Euclidean space to be constant. Their analysis was later improved by Großwendt and Röglin [44] who show approximation guarantee $O(1)$ under the assumption that the dimension is constant. In the third part of this chapter we give a simplified proof of the analysis in [44] which also yields slightly better approximation factors.

3.1 The Price of Hierarchy

Since optimal clusterings are generally not hierarchically compatible, there is usually no hierarchical clustering with approximation guarantee $\alpha = 1$. We have to accept that the restriction on hierarchically compatible clusterings comes with an unavoidable increase in the cost compared to an optimal solution.

Definition 3.1.1. *For* $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}, \mathsf{drad}\}$ *the* price of hierarchy $\rho_{\mathsf{cost}} \geq 1$ *is defined as follows.*

1. *For every instance* $(\mathcal{X}, \mathcal{P}, d)$*, there exists a hierarchical clustering* $\mathscr{H}$ *of* $\mathcal{P}$ *that is a* ρ_{cost}*-approximation with respect to* cost*.*
2. *For any* $\alpha < \rho_{\mathsf{cost}}$ *there exists an instance* $(\mathcal{X}, \mathcal{P}, d)$*, such that there is no hierarchical clustering of* $\mathcal{P}$ *that is an* α*-approximation with respect to* cost*.*

Thus ρ_{cost} is the smallest possible number such that for every clustering instance there is a hierarchical clustering with approximation guarantee ρ_{cost}. In the next two sections we focus on computing the price of hierarchy for all three objectives.

3.1.1 An Upper Bound on the Price of Hierarchy

The framework by by Lin et al. [63] can be applied to compute incremental and hierarchical solutions to a large class of minimization problems. It yields an upper bound of 4 on the price of hierarchy for the discrete radius [46]. It also yields upper bounds for the price of hierarchy for radius and diameter, which are not tight, however. We first discuss their framework in the context of hierarchical clustering for (discrete) radius and diameter. In the second part we then present an improved version of their algorithm for radius and diameter that yields the following better upper bound on the price of hierarchy for radius and diameter.

Theorem 3.1.2. *For* $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}\}$ *we have* $\rho_{\mathsf{cost}} \leq 3 + 2\sqrt{2} \approx 5.828$.

First we introduce the notion of a *hierarchical sequence*, which is a relaxation of a hierarchical clustering in the sense that it does not have to contain a k-clustering for every $1 \leq k \leq |\mathcal{P}|$.

Definition 3.1.3. *Given an instance* $(\mathcal{X}, \mathcal{P}, d)$*, with* $n = |\mathcal{P}|$*. We call a sequence* $\mathscr{C} = (\mathcal{C}^{(t)}, \dots, \mathcal{C}^{(1)})$ *of clusterings a hierarchical sequence if it satisfies*

Algorithm 4: (Lin et al. [63])

Input : Clustering instance $(\mathcal{X}, \mathcal{P}, d)$, with $d(x, y) > 2$ for all $x, y \in \mathcal{P}$, optimal clusterings $\mathcal{O}_{|\mathcal{P}|}, \ldots, \mathcal{O}_1$ of $\mathcal{P}$ with respect to cost

Output: A hierarchical clustering of $\mathcal{P}$

```
Set Δ = cost(O_1), t = ⌈log_{2γ}(Δ)⌉ + 1 and C^(t) = O_{|P|}
for i = t − 1 to 1 do
    Let 1 ≤ n_i ≤ |P| be the smallest number such that
      cost(O_{n_i}) ∈ ((2γ)^{t−i−1}, (2γ)^{t−i}]
    if such a number exists then
        set C^(i) = Augment_cost(C^(i+1), O_{n_i}, γ, δ)
    else
        set C^(i) = C^(i+1)
return h(C^(t), ..., C^(1))
```

1. $|\mathcal{C}^{(t)}| = n$ *and* $|\mathcal{C}^{(1)}| = 1$
2. *for* $1 \leq i \leq t$ *either* $\mathcal{C}^{(i-1)} = \mathcal{C}^{(i)}$ *or* $\mathcal{C}^{(i-1)}$ *is obtained from* $\mathcal{C}^{(i)}$ *by merging some of its clusters.*

Such a hierarchical sequence can be extended to a hierarchical clustering of $\mathcal{P}$ *as follows. We define the respective hierarchical clustering* $h(\mathscr{C})$ *by assigning every* $1 \leq i \leq n$ *the clustering among* $\mathcal{C}^{(t)}, \ldots, \mathcal{C}^{(1)}$ *of smallest cost and size at most* i*. We say that* $\mathscr{C}$ *is an* α*-approximation iff* $h(\mathscr{C})$ *is an* α*-approximation.*

Before we are able to define the algorithm we need one important definition from [63].

Definition 3.1.4. *Given an instance* $(\mathcal{X}, \mathcal{P}, d)$*. For* $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}, \mathsf{drad}\}$ *we say that the* (γ, δ)*-nesting property holds for reals* $\gamma, \delta \geq 0$*, if for any two clusterings* $\mathcal{C}, \mathcal{D}$ *of* $\mathcal{P}$ *with* $|\mathcal{C}| > |\mathcal{D}|$ *there exists a clustering* $\mathcal{C}'$ *with*

1. $|\mathcal{C}'| \leq |\mathcal{D}|$
2. $\mathcal{C}'$ *is hierarchically compatible with* $\mathcal{C}$ *and*
3. $\mathsf{cost}(\mathcal{C}') \leq \gamma\mathsf{cost}(\mathcal{C}) + \delta\mathsf{cost}(\mathcal{D})$.

We say that $\mathcal{C}'$ *is a* nesting *of* $\mathcal{C}$ *at* $\mathcal{D}$*. Let* $\mathsf{Augment}_{\mathsf{cost}}(\mathcal{C}, \mathcal{D}, \gamma, \delta)$ *denote the subroutine that computes such a clustering* $\mathcal{C}'$*.*

The algorithm of Lin et al. [63] is shown as Algorithm 4. It computes a hierarchical sequence $\mathscr{C} = (\mathcal{C}^{(t)}, \ldots, \mathcal{C}^{(1)})$ of clusterings as follows. Starting with $\mathcal{C}^{(t)} = \mathcal{O}_{|\mathcal{P}|}$ the algorithm builds the i-th clustering $\mathcal{C}^{(i)}$ as nesting of $\mathcal{C}^{(i+1)}$ at an optimal clustering $\mathcal{O}_{n_i}$. This guarantees that the clusterings are hierarchically compatible.

Theorem 3.1.5 ([63])**.** *For* $\mathsf{cost} \in \{\mathsf{drad}, \mathsf{rad}, \mathsf{diam}\}$*, if the* (γ, δ)*-nesting property holds for reals* $\gamma \geq 1, \delta > 0$*, then Algorithm 4 computes a hierarchical clustering of* $\mathcal{P}$ *with approximation guarantee* $4\gamma\delta$ *with respect to* cost*.*

Algorithm 5:

Input : Step size $\alpha > 1$. Clustering instance $(\mathcal{X}, \mathcal{P}, d)$, with $d(x, y) > 2$ for all $x, y \in \mathcal{P}$, optimal clusterings $\mathcal{O}_{|\mathcal{P}|}, \dots, \mathcal{O}_1$ of $\mathcal{P}$ with respect to cost

Output: A hierarchical clustering of $\mathcal{P}$

Set $\Delta = \mathsf{cost}(\mathcal{O}_1), t = \lceil \log_\alpha(\Delta) \rceil + 1$ and $\mathcal{C}^{(t)} = \mathcal{O}_{|\mathcal{P}|}$

For all $C \in \mathcal{C}^{(t)}$ we set $\mathsf{parent}_t(C) = C$

for $i = t - 1$ **to** 1 **do**

  Let $1 \leq n_i \leq |\mathcal{P}|$ be the smallest number such that $\mathsf{cost}(\mathcal{O}_{n_i}) \in (\alpha^{t-i-1}, \alpha^{t-i}]$

  if such a number exists **then**

    For $C \in \mathcal{C}^{(i+1)}$ let $O \in \mathcal{O}_{n_i}$ be a cluster with $\mathsf{parent}_{i+1}(C) \cap O \neq \emptyset$ and set $\mathsf{Nest}_i(C) = O$

    Set $\mathcal{C}^{(i)} = \{\bigcup_{C \in \mathsf{Nest}_i^{(-1)}(O)} C \mid O \in \mathcal{O}_{n_i}\}$

    Set $\mathsf{parent}_i(\bigcup_{C \in Nest_i^{(-1)}(O)} C) = O$ for all $O \in \mathcal{O}_{n_i}$

  else

    set $\mathcal{C}^{(i)} = \mathcal{C}^{(i+1)}, \mathsf{parent}_i = \mathsf{parent}_{i+1}$

return $h((\mathcal{C}^{(t)}, \dots, \mathcal{C}^{(1)}))$

Großwendt [46] proved the existence of such a nesting property for $\mathsf{diam}, \mathsf{rad}$, and drad.

Lemma 3.1.6 ([46])**.** *For* $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}\}$ *there exists a* $(2, 1)$*-nesting and for* $\mathsf{cost} = \mathsf{drad}$ *there exists a* $(1, 1)$*-nesting.*

In combination with Theorem 3.1.5 this yields $\rho_{\mathsf{drad}} \leq 4$. However, for the other two objectives we obtain an upper bound of only 8. We improve Algorithm 4 to obtain the claimed upper bound of $3 + 2\sqrt{2}$.

In the definition of the (γ, δ)-nesting property we require a nesting of $\mathcal{C}$ at $\mathcal{D}$ for arbitrary clusterings $\mathcal{C}, \mathcal{D}$ with $|\mathcal{C}| > |\mathcal{D}|$. However, in Algorithm 4 we know more about the structure of $\mathcal{C}$. This clustering is obtained by repeatedly nesting at optimal clusterings of increasing cost. In Algorithm 5 we define a nesting subroutine for this type of clusterings that eventually leads to a better approximation guarantee.

The main difference between Algorithm 4 and Algorithm 5 is the replacement of the function $\mathsf{Augment}_{\mathsf{cost}}(\mathcal{C}_{i+1}, \mathcal{O}_{n_i}, \gamma, \delta)$, which computes the nesting of $\mathcal{C}^{(i+1)}$ at $\mathcal{O}_{n_i}$, by a more explicit approach to compute such a nesting. We use the fact that $\mathcal{C}^{(i+1)}$ is obtained by a nesting at $\mathcal{O}_{n_{i+1}}$. This is reflected in the function parent_{i+1} which assigns every cluster in $\mathcal{C}^{(i+1)}$ a cluster from $\mathcal{O}_{n_{i+1}}$. In iteration i we then use the $(i+1)$-st parent function to determine which clusters of $\mathcal{C}^{(i+1)}$ will be merged to obtain $\mathcal{C}^{(i)}$. We are allowed to merge clusters $C, D \in \mathcal{C}^{(i+1)}$ if there is a cluster $O \in \mathcal{O}_{n_i}$ which has a non-empty intersection with both $\mathsf{parent}_{i+1}(C)$ and $\mathsf{parent}_{i+1}(D)$. The parent of the merged cluster in $\mathcal{C}^{(i)}$ is then set to O.

Lemma 3.1.7. *For* $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}\}$ *and any* $\alpha > 1$ *Algorithm 5 computes a hierarchical clustering with approximation guarantee* $\alpha\left(\frac{2}{\alpha - 1} + 1\right)$.

Proof. Let n denote the cardinality of $\mathcal{P}$. Notice first that $(\mathcal{C}^{(t)}, \dots, \mathcal{C}^{(1)})$ is indeed a hierarchical sequence. The first property of a hierarchical sequence is satisfied: We define

$\mathcal{C}^{(t)} = \mathcal{O}_n$ and since $\mathsf{cost}(\mathcal{O}_1) = \Delta \in (\alpha^{t-2}, \alpha^{t-1}]$ we obtain $|\mathcal{C}^{(1)}| \leq n_1 = 1$. The second property is satisfied since $\mathcal{C}^{(i)}$ either equals $\mathcal{C}^{(i+1)}$ or is obtained by merging clusters from $\mathcal{C}^{(i+1)}$. Thus Algorithm 5 indeed computes a hierarchical clustering.

Diameter ($\mathsf{cost} = \mathsf{diam}$): Let $1 \leq i \leq t$. We claim

1. for every cluster $C \in \mathcal{C}^{(i)}$ and every point $p \in \mathsf{parent}_i(C)$ that $\max_{q \in C} d(p,q) \leq \sum_{l=1}^{t-i} \alpha^l$,
2. that $\mathsf{diam}(\mathcal{C}^{(i)}) \leq \alpha^{t-i} + 2\sum_{l=1}^{t-i-1} \alpha^l$.

We prove this by induction over i, starting with $i = t$ in decreasing order. Observe that $\mathcal{C}^{(t)}$ consists only of clusters of size one so these claims are true for $i = t$.

Let $1 \leq i \leq t-1$. If $\mathcal{C}^{(i)} = \mathcal{C}^{(i+1)}$ both claims are true by induction hypothesis. Thus we assume from now on that $\mathcal{C}^{(i)} \neq \mathcal{C}^{(i+1)}$. For the first claim, we fix a cluster $C \in C^{(i)}$ and two points $p \in \mathsf{parent}_i(C)$ and $q \in C$. Let $D \in \mathcal{C}^{(i+1)}$ be the cluster which contains q. Since $\mathcal{C}^{(i)}$ is obtained by merging clusters from $\mathcal{C}^{(i+1)}$, we know that $D \subset C$ and thus $\mathsf{parent}_{i+1}(D) \cap \mathsf{parent}_i(C) \neq \emptyset$. Let $x \in \mathsf{parent}_{i+1}(D) \cap \mathsf{parent}_i(C)$. By the induction hypothesis

$$d(x,q) \leq \max_{y \in D} d(x,y) \leq \sum_{l=1}^{t-i-1} \alpha^l.$$

Since p and x both lie in $\mathsf{parent}_i(C)$ we obtain $d(p,x) \leq \mathsf{diam}(\mathcal{O}_{n_i}) \leq \alpha^{t-i}$. Using the triangle inequality we conclude

$$d(p,q) \leq d(p,x) + d(x,q) \leq \sum_{l=1}^{t-i} \alpha^l.$$

For the second claim we again fix a cluster $C \in \mathcal{C}^{(i)}$ and two points $p, q \in C$. Let $B, D \in \mathcal{C}^{(i+1)}$ such that $p \in B$ and $q \in D$. Observe that $B \cup D \subset C$ and thus $\mathsf{parent}_{i+1}(B) \cap \mathsf{parent}_i(C) \neq \emptyset \neq \mathsf{parent}_{i+1}(D) \cap \mathsf{parent}_i(C)$. Let $x_p \in \mathsf{parent}_{i+1}(B) \cap \mathsf{parent}_i(C)$ and $x_q \in \mathsf{parent}_{i+1}(D) \cap \mathsf{parent}_i(C)$. Since x_p and x_q both lie in $\mathsf{parent}_i(C)$ we obtain $d(x_p, x_q) \leq \mathsf{diam}(\mathcal{O}_{n_i}) \leq \alpha^{t-i}$. We apply the triangle inequality and the induction hypothesis of the first claim to obtain

$$d(p,q) \leq d(p,x_p) + d(x_p,x_q) + d(x_q,q) \leq \alpha^{t-i} + 2\sum_{l=1}^{t-i-1} \alpha^l.$$

Radius ($\mathsf{cost} = \mathsf{rad}$): Let $1 \leq i \leq t$. We claim that for every cluster $C \in \mathcal{C}^{(i)}$ and the center c of the cluster $\mathsf{parent}_i(C)$, it holds that $\max_{q \in C} d(c,q) \leq \alpha^{t-i} + 2\sum_{l=1}^{t-i-1} \alpha^l$. Notice that this immediately implies

$$\mathsf{rad}(\mathcal{C}^{(i)}) \leq \alpha^{t-i} + 2\sum_{l=1}^{t-i-1} \alpha^l.$$

We prove this by induction over i. Observe that $\mathcal{C}^{(t)}$ consists only of clusters of size one. So this claim is true for $i = t$. Let $1 \leq i \leq t-1$. If $\mathcal{C}^{(i)} = \mathcal{C}^{(i+1)}$ the claim is true by induction hypothesis. Thus we assume from now on that $\mathcal{C}^{(i)} \neq \mathcal{C}^{(i+1)}$. We fix a cluster $C \in C^{(i)}$, a point $q \in C$ and denote by c the center of $\mathsf{parent}_i(C)$. Let $D \in \mathcal{C}^{(i+1)}$ be the cluster which contains q. Since $\mathcal{C}^{(i)}$ is obtained by merging clusters from $\mathcal{C}^{(i+1)}$, we know

that $D \subset C$ and thus $\mathsf{parent}_{i+1}(D) \cap \mathsf{parent}_i(C) \neq \emptyset$. Let $x \in \mathsf{parent}_{i+1}(D) \cap \mathsf{parent}_i(C)$. By induction hypothesis the following holds for the center d of $\mathsf{parent}_{i+1}(D)$:

$$\max_{v \in D} d(d, v) \leq \alpha^{t-i-1} + 2 \sum_{l=1}^{t-i-2} \alpha^l$$

Together with the triangle inequality this implies

$$d(x, q) \leq d(x, d) + d(d, q) \leq \mathsf{rad}(\mathcal{O}_{n_{i+1}}) + \alpha^{t-i-1} + 2 \sum_{l=1}^{t-i-2} \alpha^l \leq 2 \sum_{l=1}^{t-i-1} \alpha^l.$$

This yields the claim, as

$$d(c, q) \leq d(c, x) + d(x, q) \leq \mathsf{rad}(\mathcal{O}_{n_i}) + 2 \sum_{l=1}^{t-i-1} \alpha^l \leq \alpha^{t-i} + 2 \sum_{l=1}^{t-i-1} \alpha^l.$$

Finally we can bound the approximation factor for both radius and diameter. Let $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}\}$. Since $d(x, y) > 2$ for all $x, y \in \mathcal{P}$ we get that $\mathsf{cost}(\mathcal{O}_{n-1}) > 1$. Thus for every $1 \leq m < n$ there is $1 \leq i \leq t-1$ such that $\mathsf{cost}(\mathcal{O}_m) \in (\alpha^{t-i-1}, \alpha^{t-i}]$. Thus the clustering $h((\mathcal{C}^{(t)}, \ldots, \mathcal{C}^{(1)}))$ is an $\alpha\left(\frac{2}{\alpha-1} + 1\right)$-approximation iff for all $1 \leq i \leq t$

$$\mathsf{cost}(\mathcal{C}^{(i)}) \leq \alpha\left(\frac{2}{\alpha - 1} + 1\right)\mathsf{cost}(\mathcal{O})$$

for all optimal clusterings $\mathcal{O}$ with $\mathsf{cost}(\mathcal{O}) \in (\alpha^{t-i-1}, \alpha^{t-i}]$. We obtain

$$\begin{aligned}\mathsf{cost}(\mathcal{C}^{(i)}) &\leq \alpha^{t-i} + 2 \sum_{l=1}^{t-i-1} \alpha^l < \alpha^{t-i} + 2 \cdot \frac{\alpha^{t-i}}{\alpha - 1} \\ &= \alpha^{t-i}\left(\frac{2}{\alpha - 1} + 1\right) \leq \alpha\left(\frac{2}{\alpha - 1} + 1\right)\mathsf{cost}(\mathcal{O}).\end{aligned}$$

□

Theorem 3.1.2. *For* $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}\}$ *we have* $\rho_{\mathsf{cost}} \leq 3 + 2\sqrt{2} \approx 5.828$.

Proof. Let $(\mathcal{X}, \mathcal{P}, d)$ be a clustering instance. We can assume without loss of generality that $d(x, y) > 2$ for all $x, y \in \mathcal{P}$, otherwise we scale the metric d accordingly. For $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}\}$ we then use Algorithm 5 with $\alpha = 1 + \sqrt{2}$ to compute a hierarchical clustering. By Lemma 3.1.7 we obtain a hierarchical clustering that is an $3 + 2\sqrt{2}$ approximation and thus $\rho_{\mathsf{cost}} \leq 3 + 2\sqrt{2}$. □

3.1.2 A Lower Bound on the Price of Hierarchy

The most challenging part is to improve the lower bounds on the price of hierarchy for diameter, radius, and discrete radius.

Theorem 3.1.8. *For* $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}\}$ *we have* $\rho_{\mathsf{cost}} \geq 3 + 2\sqrt{2}$ *and for* $\mathsf{cost} = \mathsf{drad}$ *we have* $\rho_{\mathsf{cost}} \geq 4$.

There is already existing work in this area by Das and Kenyon-Mathieu [37] for the diameter and Großwendt [46] for the radius. Both show a lower bound of 2 for the respective objective. To improve upon these results we have to construct much more complex instances which differ significantly from those in [37, 46].

For every $\epsilon > 0$ we will construct a clustering instance $(\mathcal{X}, \mathcal{P}, d)$ such that for any hierarchical clustering $\mathscr{H} = (\mathcal{H}_{|\mathcal{P}|}, \dots, \mathcal{H}_1)$ of $\mathcal{P}$ there is $1 \leq i \leq |\mathcal{P}|$ such that $\mathsf{cost}(\mathcal{H}_i) \geq \alpha \cdot \mathsf{cost}(\mathcal{O}_i)$, where $\mathcal{O}_i$ is an optimal i-clustering of $\mathcal{P}$ with respect to cost and $\alpha = (3 + 2\sqrt{2} - \epsilon)$ for $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}\}$ and $\alpha = 4 - \epsilon$ for $\mathsf{cost} = \mathsf{drad}$.

The proof is divided in three parts. First we introduce the clustering instance $(\mathcal{X}, \mathcal{P}, d)$ and determine its optimal clusterings. In the second part we develop the notion of a *bad* cluster. We prove that any hierarchical clustering contains such bad clusters and develop a lower bound on their cost. In the third part we compare the lower bound to the cost of optimal clusterings and prove Theorem 3.1.8.

Definition of the Clustering Instance

For $n \in \mathbb{N}$ we denote by $[n]$ the set of numbers from 1 to n. Let $k \in \mathbb{N}$ and $\Gamma = k + 1$. For $0 \leq \ell \leq k$ we define point sets $\mathcal{Q}_\ell$ and $\mathcal{P}_\ell$ recursively as follows

1. For $\ell = 0$ let $\mathcal{P}_0 = \mathcal{Q}_0 = [1]$ and denote by N_0 the cardinality of $\mathcal{P}_0$.
2. For $\ell > 0$ let $\mathcal{Q}_\ell = [\Gamma \cdot N_{\ell-1}]^{N_{\ell-1}}$ and $\mathcal{P}_\ell = \prod_{i=0}^{\ell} \mathcal{Q}_i$. Furthermore set $N_\ell = |\mathcal{P}_\ell|$.

Moreover let $\phi_\ell \colon \mathcal{P}_\ell \to [N_\ell]$ be a bijection for $0 \leq \ell \leq k$.

We refer to a point $X \in \mathcal{P}_k$ as a matrix with $k + 1$ rows and $N_{\ell-1}$ entries in the ℓ-th row. Thus we write

$$X = (x_{01} \mid \dots \mid x_{\ell 1}, \dots, x_{\ell N_{\ell-1}} \mid \dots \mid x_{k1}, \dots, x_{kN_{k-1}}).$$

Let $X_\ell = (x_{\ell 1}, \dots, x_{\ell N_{\ell-1}}) \in \mathcal{Q}_\ell$ for $0 \leq \ell \leq k$. For a shorter representation we can replace the ℓ-th row directly by X_ℓ and for $0 \leq i \leq j \leq k$ we can replace the i-th up to j-th row by $X_{[i:j]} = (X_i \mid \dots \mid X_j)$.

Let $X \in \mathcal{P}_k$ and $1 \leq \ell \leq k$. Notice that $X_{[0:\ell-1]} \in \mathcal{P}_{\ell-1}$ and let $m = \phi_{\ell-1}(X_{[0:\ell-1]})$, we define

$$A_\ell^X = \{(X_{[0:\ell-1]} \mid x_{\ell 1}, \dots, x_{\ell m-1}, \star, x_{\ell m+1}, \dots, x_{\ell N_{\ell-1}} \mid X_{[\ell+1:k]}) \mid \star \in [\Gamma \cdot N_{\ell-1}]\}.$$

Thus all coordinates of points in A_ℓ^X are fixed and agree with those of X except one which is variable. Here $X_{[0:\ell-1]}$ serves as prefix which indicates through $\phi_{\ell-1}$ which coordinate of X_ℓ can be changed.

We define $\mathcal{A}_\ell = \{A_\ell^X \mid X \in \mathcal{P}_k\}$ as the set containing all subsets of this form. It is clear that $\mathcal{A}_\ell$ is a partition of $\mathcal{P}_k$ and that it contains only sets of size $\Gamma \cdot N_{\ell-1}$. Furthermore we set $\mathcal{A}_0 = \{\{X\} \mid X \in \mathcal{P}_k\}$.

Example 3.1.9. *If we perform the first three steps of the construction we get* $\mathcal{Q}_0 = [1], \mathcal{Q}_1 = [\Gamma], Q_2 = [\Gamma^2]^\Gamma$ *and*

$$\mathcal{P}_1 = \{(1 \mid x_{11}) \mid x_{11} \in [\Gamma]\},$$
$$\mathcal{P}_2 = \{(1 \mid x_{11} \mid x_{21}, \dots, x_{2\Gamma}) \mid x_{11} \in [\Gamma], x_{2i} \in [\Gamma^2] \text{ for } 1 \leq i \leq \Gamma\}$$

Since ϕ_0 is a map between two sets of cardinality one this map is always unique. Now suppose that we picked ϕ_1 such that $\phi_1((x_{01} \mid x_{11})) = x_{11}$ for all $(x_{01} \mid x_{11}) \in \mathcal{P}_1$. Then the partition $\mathcal{A}_1$ consists of the sets

$$\{(1 \mid \star \mid x_{21}, \dots, x_{2\Gamma}) \mid \star \in [\Gamma]\}$$

with $x_{2i} \in [\Gamma^2]$ for all $1 \leq i \leq \Gamma$. The partition $\mathcal{A}_2$ consists of the sets

$$\{(1 \mid x_{11} \mid x_{21}, \dots, x_{2x_{11}-1}, \star, x_{2x_{11}+1}, \dots, x_{2\Gamma}) \mid \star \in [\Gamma^2]\}$$

with $x_{11} \in [\Gamma]$ and $x_{2i} \in [\Gamma^2]$ for all $1 \leq i \leq \Gamma$ with $i \neq x_{11}$.

Let $G = (V, E, w)$ denote the weighted hyper-graph with $V = \mathcal{P}_k$ and $E = \bigcup_{i=1}^{k} \mathcal{A}_i$. The weight of a hyper-edge $e \in E$ is set to ℓ iff $e \in \mathcal{A}_\ell$. For $0 \leq \ell \leq k$, the sub-graph $G_\ell = (V_\ell, E_\ell, w_\ell)$ is given by $V_\ell = \mathcal{P}_k, E_\ell = \bigcup_{i=0}^{\ell} \mathcal{A}_i$ and $w_\ell = w_{|E_\ell}$.

We extend G to a hyper-graph $H = (V', E', w')$ as follows. Let $V' = V \cup \bigcup_{i=0}^{k} \{v_A \mid A \in \mathcal{A}_i\}$ and $E' = E \cup \bigcup_{i=0}^{k} \{\{v, v_A\} \mid A \in \mathcal{A}_i, v \in A\}$. Thus H contains one vertex for every $A \in \bigcup_{i=0}^{k} \mathcal{A}_i$ and this vertex is connected by edges to every vertex $v \in A$. For $e \in E$ we set $w'(e) = w(e)$ and for $e = \{v, v_A\}$ for some $A \in \mathcal{A}_\ell$ and $v \in A$ we set $w'(e) = \ell/2$.

The clustering instance $(\mathcal{X}, \mathcal{P}, d)$ is given by $\mathcal{X} = V', \mathcal{P} = V$, and d as the shortest path metric on H. Observe that the extension of G to H is only necessary for the lower bound for the radius but not for the diameter and the discrete radius. This is because the additional points $V' \setminus V$ do not belong to $\mathcal{P}$ and are hence irrelevant for the clustering instance for the diameter and discrete radius. In the lower bound for the radius they will be used as centers, however.

Lemma 3.1.10. *Let $p, q \in V$, then $d(p, q)$ is the length of a shortest path between p and q in G.*

Proof. By definition $d(p, q)$ is the length of a shortest path between p and q in H. Suppose the shortest path contains a vertex v_A for some $A \in \bigcup_{i=0}^{k} \mathcal{A}_i$ with $v \in A$ as predecessor and $w \in A$ as ancestor. Since v and w are connected in H by the hyper-edge A we can delete v_A from the path and the length of the path does not change. The resulting path is also a path in G, so $d(p, q)$ is also the length of a shortest path between p and q in G. □

Next we state some structural properties of the graph G and the clustering instance $(\mathcal{X}, \mathcal{P}, d)$. To establish a lower bound on the approximation factor of a hierarchical clustering we first focus on the optimal clusterings of the instance $(\mathcal{X}, \mathcal{P}, d)$. One can already guess that $\mathcal{A}_\ell$ is an optimal clustering with $\frac{N_k}{\Gamma N_{\ell-1}}$ clusters with respect to $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}, \mathsf{drad}\}$ and we will prove this in this section. First we need the following statement about the connected components of G_ℓ.

Lemma 3.1.11. *The vertex set of every connected component in G_ℓ has cardinality N_ℓ and is of the form $V_\ell^X = \{(X' \mid X) \mid X' \in \mathcal{P}_\ell\}$ for a given $X = (X_{\ell+1} \mid \dots \mid X_k) \in \prod_{i=\ell+1}^{k} \mathcal{Q}_i$.*

Proof. Notice that $|V_\ell^X| = N_\ell$ and that $\{V_\ell^X \mid X \in \prod_{i=\ell+1}^{k} \mathcal{Q}_i\}$ is a partition of V. Furthermore since $E_\ell = \bigcup_{i=0}^{\ell} \mathcal{A}_i$ any edge $e \in E_\ell$ is either completely contained in or disjoint to V_ℓ^X.

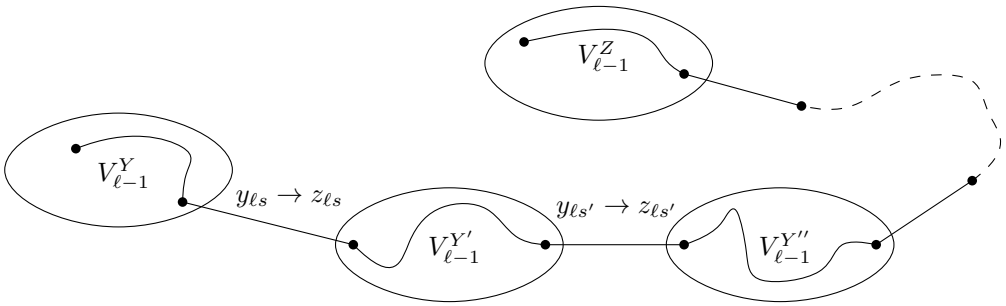

Figure 3.2: Here we see the construction of the path. It corresponds to changing the coordinates of Y successively until they match Z. We use an edge in $\mathcal{A}_\ell$ to change y_{ls} to z_{ls}, next we change $y_{ls'}$ to $z_{ls'}$ and proceed like this until we obtain Z. The respective edges are then connected to a path from $V_{\ell-1}^X$ to $V_{\ell-1}^Z$.

It is left to show that V_ℓ^X is connected. We prove this via induction over ℓ. For $\ell = 0$ this is clear because $|V_0^X| = 1$. For $\ell > 0$ let $Y = (Y_\ell \mid X), Z = (Z_\ell \mid X) \in \prod_{i=\ell}^k \mathcal{Q}_i$. By the induction hypothesis we know that the sets $V_{\ell-1}^Y, V_{\ell-1}^Z$ are connected. To prove that V_ℓ^X is connected it is sufficient to show that there is a path from a point in $V_{\ell-1}^Y$ to a point in $V_{\ell-1}^Z$. We show this claim by induction over the number m of coordinates in which Y and Z differ. For $m = 0$ there is nothing to show. If $m > 0$ pick $1 \leq s \leq N_{\ell-1}$ such that $y_{\ell s} \neq z_{\ell s}$ and let $P = \phi_{\ell-1}^{-1}(s) \in \prod_{i=0}^{\ell-1} \mathcal{Q}_i$. Consider the point $(P \mid Y_\ell \mid X)$ which is contained in $V_{\ell-1}^Y$. This point is also contained in the set

$$\{(P \mid y_{\ell 1}, \ldots, y_{\ell s-1}, \star, y_{\ell s+1}, \ldots, y_{\ell N_{\ell-1}} \mid X) \mid \star \in [\Gamma \cdot N_{\ell-1}]\} \in E_\ell.$$

Thus there is an edge in G_ℓ connecting a point in $V_{\ell-1}^Y$ to a point in $V_{\ell-1}^{Y'}$ with $Y' = (y_{\ell 1}, \ldots, y_{\ell s-1}, z_{\ell s}, y_{\ell s+1}, \ldots, y_{N_{\ell-1}} \mid X)$. Now Y' and Z differ in $m-1$ coordinates, thus there is a path between two points in $V_{\ell-1}^{Y'}$ and $V_{\ell-1}^Z$ by induction hypothesis. If we combine this with the induction hypothesis that $V_{\ell-1}^{Y'}$ is connected this yields the claim (see Figure 3.2 for an illustration). □

Lemma 3.1.12. *Any clustering of $(\mathcal{X}, \mathcal{P}, d)$ with fewer than $\frac{N_k}{N_{\ell-1}}$ clusters costs at least ℓ if $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{drad}\}$ and $\ell/2$ if $\mathsf{cost} = \mathsf{rad}$.*

Proof. The shortest path in G between any two points which lie in different connected components of $G_{\ell-1}$ must contain an edge of weight $\geq \ell$. Thus any set of points $M \subset V$ which is disconnected in $G_{\ell-1}$ has diameter $\geq \ell$. Remember that the discrete radius of M is given by $\mathsf{drad}(M) = \min_{c \in M} \max_{p \in M} d(p, c)$. For every possible choice of $c \in M$ there exists a point $p \in M$ which is not in the same connected component of $G_{\ell-1}$ as c, thus $d(c, p) \geq \ell$ and therefore $\mathsf{drad}(M) \geq \ell$ and $\mathsf{rad}(M) \geq \mathsf{diam}(M)/2 \geq \ell/2$.

We conclude that if $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{drad}\}$ any cluster of cost smaller than ℓ is contained in one of the sets $V_{\ell-1}^X$ for some $X \in \prod_{i=\ell}^k \mathcal{Q}_i$ by Lemma 3.1.11 and any clustering with fewer than $\left|\prod_{i=\ell}^k \mathcal{Q}_i\right|$ clusters costs at least ℓ. By the same argument if $\mathsf{cost} = \mathsf{rad}$ any cluster of cost smaller than $\ell/2$ is contained in one of the sets $V_{\ell-1}^X$ for some $X \in \prod_{i=\ell}^k \mathcal{Q}_i$ by Lemma 3.1.11 and any clustering with fewer than $\left|\prod_{i=\ell}^k \mathcal{Q}_i\right|$ clusters costs at least

$\ell/2$. Since

$$\left|\prod_{i=\ell}^{k} \mathcal{Q}_i\right| = \frac{\left|\prod_{i=0}^{k} \mathcal{Q}_i\right|}{\left|\prod_{i=0}^{\ell-1} \mathcal{Q}_i\right|} = \frac{N_k}{N_{\ell-1}}$$

this proves the lemma. □

Corollary 3.1.13. *For $1 \leq \ell \leq k$ and $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}, \mathsf{drad}\}$ the clustering $\mathcal{A}_\ell$ is an optimal $\frac{N_k}{\Gamma N_{\ell-1}}$-clustering for the instance $(\mathcal{X}, \mathcal{P}, d)$. Furthermore $\mathsf{diam}(\mathcal{A}_\ell) = \mathsf{drad}(\mathcal{A}_\ell) = \ell$ and $\mathsf{rad}(\mathcal{A}_\ell) = \ell/2$.*

Proof. If $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{drad}\}$ we obtain by definition of $(\mathcal{X}, \mathcal{P}, d)$ that $\mathsf{cost}(\mathcal{A}_\ell) \leq \ell$. If $\mathsf{cost} = \mathsf{rad}$ we obtain that $\mathsf{cost}(\mathcal{A}) \leq \ell/2$ by picking $v_A \in \mathcal{X} \backslash \mathcal{P}$ as center for $A \in \mathcal{A}_\ell$. On the other hand $|\mathcal{A}_\ell| = \frac{N_k}{\Gamma N_{\ell-1}} < \frac{N_k}{N_{\ell-1}}$ and thus $\mathsf{cost}(\mathcal{A}_\ell) \geq \ell$ if $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{drad}\}$ and $\mathsf{cost}(\mathcal{A}_\ell) \geq \ell/2$ for $\mathsf{cost} = \mathsf{rad}$ by Lemma 3.1.12. □

Characterization of Hierarchical Clusterings

Let from now on $\mathscr{H} = (\mathcal{H}_{N_k}, \ldots, \mathcal{H}_1)$ denote a hierarchical clustering of $(\mathcal{X}, \mathcal{P}, d)$. We introduce the notion of *bad clusters* in $\mathcal{H}_{\frac{N_k}{\Gamma N_{\ell-1}}}$ which are clusters whose cost increases repeatedly, as we will see later. In this section we prove the existence of such clusters in $\mathscr{H}$ and we give a lower bound on their cost.

Definition 3.1.14. *We call all clusters $C \in \mathcal{H}_{N_k}$ bad at time 0 and denote by $\mathsf{Ker}_0(C) = C$ the kernel of C at time 0 and set $\mathsf{Bad}(0) = \mathcal{H}_{N_k}$.*

For $1 \leq \ell \leq k$ we say that a cluster $C \in \mathcal{H}_{\frac{N_k}{\Gamma N_{\ell-1}}}$ is anchored at $\ell \leq \ell' \leq k$ if the set $\bigcup_{D \in \mathsf{Bad}(\ell-1):\, D \subset C} \mathsf{Ker}_{\ell-1}(D)$ is

1. *connected in $G_{\ell'}$,*
2. *disconnected in $G_{\ell'-1}$.*

We call C bad at time ℓ if C is anchored at some $\ell' \geq \ell$. We denote by $\mathsf{Bad}(\ell) \subset \mathcal{H}_{\frac{N_k}{\Gamma N_{\ell-1}}}$ the set of all bad clusters at time ℓ. If C is bad we define the kernel of C as the union of all kernels of bad clusters at time $\ell - 1$ contained in C, i.e.,

$$\mathsf{Ker}_\ell(C) = \bigcup_{D \in \mathsf{Bad}(\ell-1):\, D \subset C} \mathsf{Ker}_{\ell-1}(D).$$

All clusters in $\mathcal{H}_{\frac{N_k}{\Gamma N_{\ell-1}}} \backslash \mathsf{Bad}(\ell)$ are called good.

The example in Figure 3.3 shows that a bad cluster at time ℓ can contain clusters which are good at time $\ell - 1$. However we are only interested in points that are contained exclusively in bad clusters at any time $t < \ell$. The set $\mathsf{Ker}_\ell(C)$ contains exactly such points.

We will use two crucial properties to prove the final lower bound on the approximation factor of any hierarchical clustering $\mathscr{H}$ of $(\mathcal{X}, \mathcal{P}, d)$. We first observe that bad clusters exist in $\mathscr{H}$ for every time-step $1 \leq \ell \leq k$ and second that these clusters have a large cost compared to the optimal clustering.

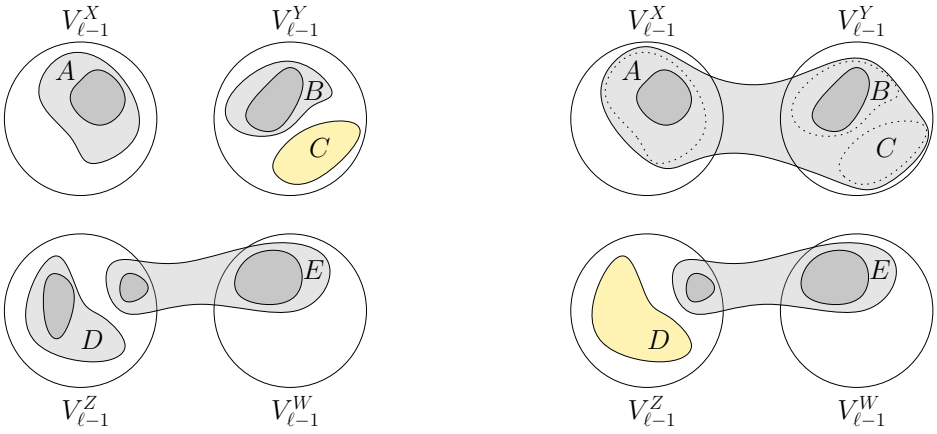

Figure 3.3: An illustration of the evolution of good and bad clusters: In the example, we see five clusters at time $\ell - 1$. The clusters A, B, D, E are assumed to be bad, with their kernels depicted in dark gray, while C is assumed to be a good cluster. At time ℓ, clusters A, B and C are merged. The resulting cluster is bad because the kernels of A and B lie in different connected components of $G_{\ell-1}$. Clusters D and E are still present at time ℓ, but now D is a good cluster since its kernel is completely contained in $V^Z_{\ell-1}$, while E is still bad, since its kernel is disconnected in $G_{\ell-1}$.

Lemma 3.1.15. *Let C be a good cluster at time $1 \leq \ell \leq k$ and*

$$W = \bigcup_{D \in \mathsf{Bad}(\ell-1)\colon D \subset C} \mathsf{Ker}_{\ell-1}(D),$$

then W is connected in $G_{\ell-1}$ and thus $|W| \leq N_{\ell-1}$.

Proof. Suppose W is disconnected in $G_{\ell-1}$. Since $G_k = G$ is connected, there must be a time $\ell' \geq \ell$ such that W is connected in $G_{\ell'}$ and disconnected in $G_{\ell'-1}$. But then C is a bad cluster at time ℓ which is anchored at ℓ' in contradiction to our assumption. Thus W is connected in $G_{\ell-1}$. By Lemma 3.1.11 we know that every connected component in $G_{\ell-1}$ is of size $N_{\ell-1}$. □

Lemma 3.1.16. *For all $0 \leq \ell \leq k$ we have $\sum_{C \in \mathsf{Bad}(\ell)} |\mathsf{Ker}_\ell(C)| \geq \frac{\Gamma-\ell}{\Gamma} N_k$.*

Proof. We prove this via induction over ℓ. For $\ell = 0$ this is clear since $\bigcup_{C \in \mathsf{Bad}(0)} \mathsf{Ker}_0(C) = \mathcal{P}_k$.

Now suppose that $\ell > 0$ and that

$$\sum_{C \in \mathsf{Bad}(\ell)} |\mathsf{Ker}_\ell(C)| < \frac{\Gamma - \ell}{\Gamma} N_k.$$

By induction hypothesis we know that

$$\sum_{C \in \mathsf{Bad}(\ell-1)} |\mathsf{Ker}_{\ell-1}(C)| \geq \frac{\Gamma - \ell + 1}{\Gamma} N_k.$$

Thus the number of points which are in the kernel of a bad cluster at time $\ell - 1$ but not at time ℓ is larger than

$$\frac{\Gamma - \ell + 1}{\Gamma} N_k - \frac{\Gamma - \ell}{\Gamma} N_k = \frac{N_k}{\Gamma}.$$

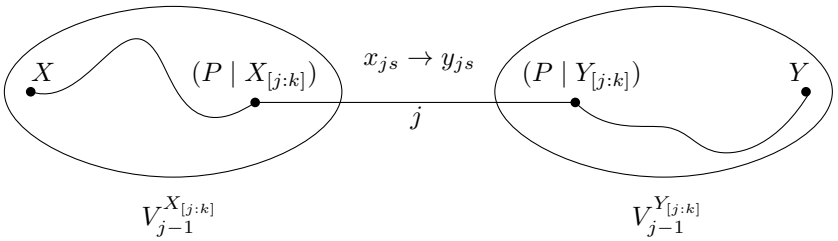

Figure 3.4: A shortest path between X and Y. It consists of two shortest paths inside the connected components of G_{j-1} and the unique edge of weight j between these components.

In other words these are points that are in the kernel of a bad cluster at time $\ell - 1$ but contained in a good cluster at time ℓ. Now we use that any good cluster at time ℓ can contain only $N_{\ell-1}$ such points by Lemma 3.1.15. Thus the number of good clusters is greater than

$$\frac{N_k}{\Gamma} \cdot \frac{1}{N_{\ell-1}} = \frac{N_k}{\Gamma N_{\ell-1}}.$$

We obtain that $\mathcal{H}_{\frac{N_k}{\Gamma N_{\ell-1}}}$ contains more than $\frac{N_k}{\Gamma N_{\ell-1}}$ clusters, which is not possible. □

Remember that $\Gamma = k + 1$, thus an immediate consequence of Lemma 3.1.16 is the existence of bad clusters at time ℓ for any $0 \leq \ell \leq k$. To prove that their (discrete) radius and diameter is indeed large we need a lower bound on the distance between two points $X, Y \in \mathcal{P}$ that lie in different connected components of G_{j-1} for some $1 \leq j \leq k$.

Suppose that the points X and Y only differ in one coordinate, i.e., there is a $1 \leq s \leq N_{j-1}$ such that $x_{js} \neq y_{js}$, while X and Y agree in all other coordinates. There is only one edge in G_j connecting $V_{j-1}^{X_{[j:k]}}$ with $V_{j-1}^{Y_{[j:k]}}$. Let $P = \phi_{j-1}^{-1}(s)$, then this edge connects the points $(P \mid X_{[j:k]})$ and $(P \mid Y_{[j:k]})$. If we connect X to $(P \mid X_{[j:k]})$ and $(P \mid Y_{[j:k]})$ to Y via a shortest path, this results in a path from X to Y, see Figure 3.4. We show that this path is indeed a shortest path between X and Y and generalize this to arbitrary X and Y which are disconnected in G_{j-1}.

Lemma 3.1.17. *Let $X, Y \in \mathcal{P}$ be two points and suppose there is $1 \leq j \leq k$ and $1 \leq s \leq N_{j-1}$ such that $x_{js} \neq y_{js}$. Let $P = \phi_{j-1}^{-1}(s) \in \prod_{i=0}^{j-1} \mathcal{Q}_i$. Then*

$$d(X, Y) \geq d\Big(X, (P \mid X_{[j:k]})\Big) + j + d\Big(Y, (P \mid Y_{[j:k]})\Big).$$

Proof. Observe that if two points in G are connected by an edge they differ in exactly one coordinate. Since $x_{js} \neq y_{js}$ any shortest path connecting X and Y must contain two consecutive points Z, Z' with $Z = (P \mid Z_j \mid \ldots \mid Z_k)$ and $Z' = (P \mid Z'_j \mid \ldots \mid Z'_k)$ such that $z_{js} = x_{js}, z'_{js} = y_{js}$ and Z agrees with Z' in all remaining coordinates. We obtain

$$d(X, Y) = d\Big(X, Z\Big) + d(Z, Z') + d\Big(Z', Y\Big) = d\Big(X, Z\Big) + j + d\Big(Z', Y\Big).$$

It is now left to show that $d(X, Z) \geq d\Big(X, (P \mid X_{[j:k]})\Big)$ and $d(Y, Z') \geq d\Big(Y, (P \mid Y_{[j:k]})\Big)$. To prove this we consider a shortest path $V^1, \ldots, V^t$ connecting $V^1 = X$ with

$V^t = Z$. Let $W^i = (V^i_{[0:j-1]} \mid X_{[j:k]})$ for $i = 1, \ldots, t$. We claim that W^i is connected to W^{i+1} by an edge in G and that $d(V^i, V^{i+1}) \geq d(W^i, W^{i+1})$ for all $1 \leq i \leq t-1$. So let $1 \leq i \leq t-1$, we know that V^i and V^{i+1} differ in exactly one coordinate. If they differ at a coordinate in row $r \geq j$ we have $W^i = W^{i+1}$ and thus the claim holds. Otherwise let $u = \phi_{r-1}(V^i_{[0:r-1]})$ then V^i and V^{i+1} satisfy $v^i_{ru} \neq v^{i+1}_{ru}$ and $d(V^i, V^{i+1}) = r$. Since $r \leq j-1$ we obtain that W^i is connected to W^{i+1} by the edge

$$\{(V^i_{[0:r-1]} \mid v^i_{r1}, \ldots, v^i_{ru-1}, \star, v^i_{ru+1}, \ldots, v^i_{rN_{r-1}} \mid W^i_{[r+1:k]}) \mid \star \in [\Gamma N_{r-1}]\},$$

which has weight r. This yields the claim.

Observe that $W^1 = X$ and $W^t = (P \mid X_{[j:k]})$ and that

$$d\Big(X, (P \mid X_{[j:k]})\Big) \leq \sum_{i=1}^{t-1} d(W^i, W^{i+1}) \leq \sum_{i=1}^{t-1} d(V^i, V^{i+1}) = d(X, Z).$$

Analogously one can show $d(Y, Z') \geq d\Big(Y, (P \mid Y_{[j:k]})\Big)$ and obtains

$$d(X, Y) = d\Big(X, Z\Big) + j + d\Big(Z', Y\Big) \geq d\Big(X, (P \mid X_{[j:k]})\Big) + j + d\Big(Y, (P \mid Y_{[j:k]})\Big).$$

□

We now define the so called *anchor set* $\mathsf{Anc}_\ell(C)$ of a bad cluster C at time ℓ. If C is anchored at ℓ' then $\mathsf{Anc}_\ell(C)$ is the union of ℓ' and the anchor set of some bad cluster $D \subset C$ at time $\ell - 1$. If we choose D appropriately the sum of anchors in $\mathsf{Anc}_\ell(C)$ is a lower bound on the discrete radius of C, as we show later. It is clear that ℓ' itself is a lower bound on the discrete radius since $\mathsf{Ker}_\ell(C)$ is disconnected in $G_{\ell'-1}$ by definition. If we additionally assume that the discrete radius of D is large, e.g., lower bounded by the sum of anchors in $\mathsf{Anc}_{\ell-1}(D)$, then it is reasonable to assume that the discrete radius of C is lower bounded by some function in ℓ' and the sum of anchors in $\mathsf{Anc}_{\ell-1}(D)$. Before proving this we give a formal definition of $\mathsf{Anc}_\ell(C)$ and how to choose D.

Definition 3.1.18. *Let $1 \leq \ell \leq k$ and C be a bad cluster at time ℓ which is anchored at $\ell' \geq \ell$. If $\ell = 1$ we define the anchor set of C as $\mathsf{Anc}_1(C) = \{\ell'\}$ and set $\mathsf{prev}(C) = \{X\}$ for some $X \in C$.*

For $\ell > 1$ we distinguish two cases.

Case 1: *C contains a bad cluster D which is bad at time $\ell - 1$ and anchored at ℓ'. We then set $\mathsf{Anc}_\ell(C) = \mathsf{Anc}_{\ell-1}(D)$ and $\mathsf{prev}(C) = D$.*

Case 2: *C does not contain such a cluster. Then let $D \subset C$ be a bad cluster at time $\ell - 1$ minimizing*

$$\sum_{a \in \mathsf{Anc}_{\ell-1}(D)} a$$

among all clusters $D' \in \mathsf{Bad}(\ell - 1)$ with $D' \subset C$. We set $\mathsf{Anc}_\ell(C) = \mathsf{Anc}_{\ell-1}(D) \cup \{\ell'\}$ and $\mathsf{prev}(C) = D$.

Observe that in Case 2 of the previous definition, the bad cluster D must be anchored at some $\ell_D < \ell'$.

Lemma 3.1.19. *Let $1 \leq \ell \leq k$ and C be a bad cluster at time ℓ. If C contains a cluster D which is bad at time $\ell - 1$ then $\mathsf{Ker}_{\ell-1}(D) \subset \mathsf{Ker}_\ell(C)$.*

Proof. Since $D \in \mathsf{Bad}(\ell-1)$ and $D \subset C$, we get

$$\mathsf{Ker}_{\ell-1}(D) \subset \bigcup_{D' \subset \mathsf{Bad}(\ell-1)\colon\, D' \subset C} \mathsf{Ker}_{\ell-1}(D') = \mathsf{Ker}_{\ell}(C).$$

□

With the help of Lemma 3.1.17 we are able to show how the discrete radius and diameter of a bad cluster, depends on the sum of anchors.

Lemma 3.1.20. *Let $1 \leq \ell \leq k$ and C be a bad cluster at time ℓ anchored at ℓ'. Then for any point $Z \in \mathcal{P}$ there is $X \in \mathsf{Ker}_\ell(C)$ such that*

$$d(Z, X) \geq \sum_{a \in \mathsf{Anc}_\ell(C)} a.$$

Proof. Let $Z \in \mathcal{P}$ and suppose that C is a bad cluster at time ℓ anchored at ℓ'. We prove the lemma via induction over ℓ. For $\ell = 1$ we know that $\mathsf{Ker}_\ell(C)$ is disconnected in $G_{\ell'-1}$ by definition. Thus there is a point $X \in \mathsf{Ker}_\ell(C)$ which is disconnected from Z in $G_{\ell'-1}$ yielding

$$d(Z, X) \geq \ell' = \sum_{a \in \mathsf{Anc}_1(C)} a.$$

Let $\ell > 1$. If $D = \mathsf{prev}(C)$ is anchored at ℓ' we apply Lemma 3.1.19 to observe that $\mathsf{Ker}_{\ell-1}(D) \subset \mathsf{Ker}_\ell(C)$. By induction hypothesis the lemma holds for D. Since $\mathsf{Anc}_\ell(C) = \mathsf{Anc}_{\ell-1}(D)$ the lemma also holds for C.

Otherwise let $D = \mathsf{prev}(C)$ be anchored at $\ell_D < \ell'$. We know that $\mathsf{Ker}_\ell(C)$ is disconnected in $G_{\ell'-1}$. On the other hand $\mathsf{Ker}_{\ell-1}(D)$ is connected in $G_{\ell'-1}$ since $\ell_D < \ell'$. Thus there is $V \in \mathsf{Ker}_\ell(C)$ which is disconnected from $\mathsf{Ker}_{\ell-1}(D)$ in $G_{\ell'-1}$. Let $E \subset C$ be the cluster at time $\ell - 1$ which contains V. Since $V \in \mathsf{Ker}_\ell(C)$ we know that E is a bad cluster at time $\ell - 1$ anchored at $\ell_E < \ell'$. We know that $\mathsf{Ker}_{\ell-1}(E)$ is connected in $G_{\ell'-1}$ and lies in a different connected component than $\mathsf{Ker}_{\ell-1}(D)$. Thus Z is disconnected from $\mathsf{Ker}_{\ell-1}(D)$ or $\mathsf{Ker}_{\ell-1}(E)$ in $G_{\ell'-1}$.

We assume without loss of generality that Z is disconnected from E in $G_{\ell'-1}$. Since $\mathsf{Ker}_{\ell-1}(E)$ is connected in $G_{\ell'-1}$ we know by Lemma 3.1.11 that $(P \mid Y_{[\ell':k]}) = (P \mid Y'_{[\ell':k]})$ for all $Y, Y' \in \mathsf{Ker}_{\ell-1}(E)$. Also by Lemma 3.1.11 there is $\ell' \leq r \leq k$ and $1 \leq s \leq N_{r-1}$ such that $z_{rs} \neq y_{rs}$ for all $Y \in \mathsf{Ker}_{\ell-1}(E)$. Let $P = \phi_{r-1}^{-1}(s)$. Thus we know by induction hypothesis that there is a point $X \in \mathsf{Ker}_{\ell-1}(E) \subset \mathsf{Ker}_\ell(C)$ with

$$d(X, (P \mid X_{[r:k]})) \geq \sum_{a \in \mathsf{Anc}_{\ell-1}(E)} a.$$

Figure 3.5 shows an exemplary path between X and Z.

We apply Lemma 3.1.17 to see that

$$\begin{aligned} d(Z, X) &\geq d\Big(Z, (P \mid Z_{[r:k]})\Big) + r + d\Big(X, (P \mid X_{[r:k]})\Big) \\ &\geq r + \sum_{a \in \mathsf{Anc}_{\ell-1}(E)} a \\ &\geq \ell' + \sum_{a \in \mathsf{Anc}_{\ell-1}(E)} a \\ &\geq \sum_{a \in \mathsf{Anc}_{\ell}(C)} a \end{aligned}$$

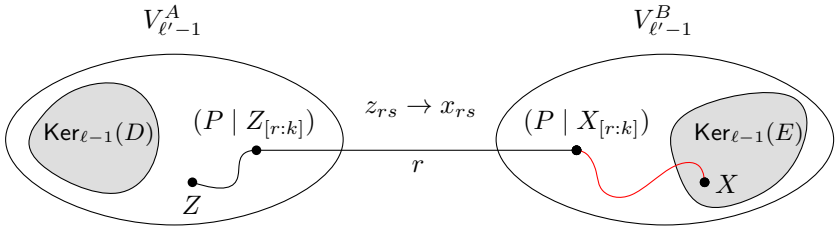

Figure 3.5: Shows the special case where $Z_{[r:k]}$ and $Y_{[r:k]}$ only differ in the rs-coordinate. The length of the red path is lower bounded by $\sum_{a \in \mathsf{Anc}_{\ell-1}(E)} a$.

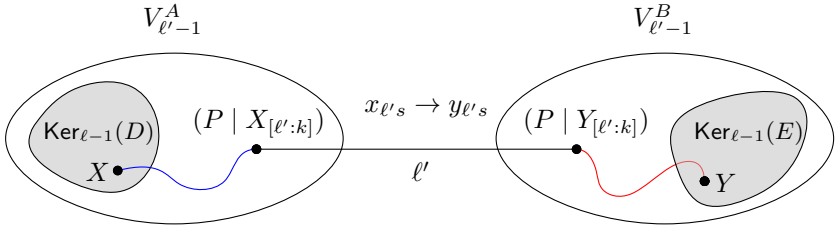

Figure 3.6: Shows the special case where $X_{[\ell':k]}$ and $Y_{[\ell':k]}$ only differ in the $\ell's$-coordinate. The length of the blue path is lower bounded by $\sum_{a \in \mathsf{Anc}_{\ell-1}(D)} a$, while the length of the red path is lower bounded by $\sum_{a \in \mathsf{Anc}_{\ell-1}(E)} a$.

Here the last inequality follows from the minimality of $\sum_{a \in \mathsf{Anc}_{\ell-1}(D)} a$ among all clusters $D' \in \mathsf{Bad}(\ell-1)$ with $D' \subset C$.

If Z is disconnected from D in $G_{\ell'-1}$ our argument still works after replacing E by D. □

Lemma 3.1.21. *Let $1 \leq \ell \leq k$ and C be a bad cluster at time ℓ anchored at ℓ'. Then there are two points $X, Y \in \mathsf{Ker}_\ell(C)$ such that*

$$d(X,Y) \geq \ell' + 2 \sum_{a \in \mathsf{Anc}_\ell(C) \setminus \{\ell'\}} a.$$

Proof. Suppose that C is a bad cluster at time ℓ anchored at ℓ'. We prove the lemma via induction over ℓ. For $\ell = 1$ we know that $\mathsf{Ker}_\ell(C)$ is disconnected in $G_{\ell'-1}$ by definition. Thus there are two points $X, Y \in \mathsf{Ker}_\ell(C)$ that are disconnected in $G_{\ell'-1}$ yielding

$$d(X,Y) \geq \ell' = \ell' + 2 \sum_{a \in \mathsf{Anc}_1(C) \setminus \{\ell'\}} a.$$

Let $\ell > 1$. If $D = \mathsf{prev}(C)$ is anchored at ℓ' we apply Lemma 3.1.19 to observe that $\mathsf{Ker}_{\ell-1}(D) \subset \mathsf{Ker}_\ell(C)$. By induction hypothesis the lemma holds for D. Since $\mathsf{Anc}_\ell(C) = \mathsf{Anc}_{\ell-1}(D)$ the lemma also holds for C.

Otherwise let $D = \mathsf{prev}(C)$ be anchored at $\ell_D < \ell'$. We know that $\mathsf{Ker}_\ell(C)$ is disconnected in $G_{\ell'-1}$ and $\mathsf{Ker}_{\ell-1}(D)$ is connected in $G_{\ell'-1}$. Thus there is $V \in \mathsf{Ker}_\ell(C)$ which is disconnected from $\mathsf{Ker}_{\ell-1}(D)$ in $G_{\ell'-1}$. Let $E \subset C$ be the cluster at time $\ell - 1$ which contains V. We know that E is a bad cluster at time $\ell - 1$ anchored at $\ell_E < \ell'$.

Furthermore $\mathsf{Ker}_{\ell-1}(E)$ is connected in $G_{\ell'-1}$ and lies in a different connected component than $\mathsf{Ker}_{\ell-1}(D)$.

Since $\mathsf{Ker}_{\ell-1}(D)$ and $\mathsf{Ker}_{\ell-1}(E)$ are disconnected in $G_{\ell'-1}$ but connected in $G_{\ell'}$, there must be $1 \leq s \leq N_{\ell'-1}$ such that for all $U \in \mathsf{Ker}_{\ell-1}(D)$ and $T \in \mathsf{Ker}_{\ell-1}(E)$ we have $u_{\ell' s} \neq t_{\ell' s}$ by Lemma 3.1.11. Let $P = \phi_{\ell'-1}^{-1}(s)$, we know by Lemma 3.1.17 that

$$d(U,T) \geq d(U,(P \mid U_{[\ell':k]})) + \ell' + d(T,(P \mid T_{[\ell':k]})).$$

Let $U \in \mathsf{Ker}_{\ell-1}(D)$ and $T \in \mathsf{Ker}_{\ell-1}(E)$. We know by Lemma 3.1.20 that for any two points $Z = (P \mid U_{[\ell':k]})$ and $Z' = (P \mid T_{[\ell':k]})$ there must be $X \in \mathsf{Ker}_{\ell-1}(D)$ and $Y \in \mathsf{Ker}_{\ell-1}(E)$ such that

$$d(X,Z) \geq \sum_{a \in \mathsf{Anc}_{\ell-1}(D)} a$$

and

$$d(Y,Z') \geq \sum_{a \in \mathsf{Anc}_{\ell-1}(E)} a.$$

We use Lemma 3.1.11 to observe that $Z = (P \mid X_{[\ell':k]})$ and $Z' = (P \mid Y_{[\ell':k]})$ because X is connected to U and Y is connected to T in $G_{\ell'-1}$. Figure 3.6 shows an exemplary path between X and Y. Thus

$$\begin{aligned} d(X,Y) &\geq d(X,(P \mid X_{[\ell':k]})) + \ell' + d(Y,(P \mid Y_{[\ell':k]})) \\ &\geq d(X,Z) + \ell' + d(Y,Z') \\ &\geq \ell' + \sum_{a \in \mathsf{Anc}_{\ell-1}(D)} a + \sum_{a \in \mathsf{Anc}_{\ell-1}(E)} a \\ &\geq \ell' + 2 \sum_{a \in \mathsf{Anc}_{\ell}(C) \setminus \{\ell'\}} a \end{aligned}$$

Here the last inequality follows from the minimality of $\sum_{a \in \mathsf{Anc}_{\ell-1}(D)} a$ among all clusters $D' \in \mathsf{Bad}(\ell-1)$ with $D' \subset C$. □

Comparison to Optimal Clusterings Our initial motivation was to construct an instance where any hierarchical clustering has a high approximation ratio. If we consider an arbitrary time $1 \leq \ell \leq k$ then the hierarchical clustering $\mathscr{H}$ on $(\mathcal{X}, \mathcal{P}, d)$ may be even optimal at time ℓ. Thus the bounds which we develop in Lemma 3.1.20 and Lemma 3.1.21 on the discrete radius and diameter of bad clusters are useless without linking the cost of a bad cluster at time ℓ to the cost of bad clusters at other time steps. Therefore we construct a sequence of clusters $C_1 \subset C_2 \ldots \subset C_k$ where C_i is a bad cluster at time i such that $\mathsf{Anc}_1(C_1) \subset \mathsf{Anc}_2(C_2) \subset \ldots \subset \mathsf{Anc}_k(C_k)$. We then show with the help of Lemma 3.1.20 and Lemma 3.1.21 that at least one of these clusters has a high discrete radius and diameter compared to the optimal cost.

Lemma 3.1.22. *Let C_k be a bad cluster at time k. For $1 \leq i \leq k-1$ we define $C_i = \mathsf{prev}(C_{i+1})$. For all $1 \leq i \leq k-1$ cluster C_i is bad at time i and one of the following two cases occurs:*

1. $\mathsf{Anc}_i(C_i) = \mathsf{Anc}_{i+1}(C_{i+1})$,
2. $\mathsf{Anc}_{i+1}(C_{i+1}) \setminus \{\ell\} = \mathsf{Anc}_i(C_i)$, *where* $\ell = \max \mathsf{Anc}_{i+1}(C_{i+1})$.

Proof. For $i = k$ cluster C_k is bad at time k by assumption. If C_{i+1} is a bad cluster at time $i+1$ then $C_i = \mathsf{prev}(C_{i+1})$ is a bad cluster at time i, by definition of prev.

Let C_i be anchored at $\ell' \geq i$ and C_{i+1} be anchored at $\ell \geq i+1$. Since $\mathsf{Ker}_i(C_i) \subset \mathsf{Ker}_{i+1}(C_{i+1})$ by Lemma 3.1.19, we know that $\ell' \leq \ell$. If $\ell' = \ell$ we obtain by Definition 3.1.18, that $\mathsf{Anc}_i(C_i) = \mathsf{Anc}_{i+1}(C_{i+1})$, so the lemma holds in this case.

If $\ell' < \ell$ we know by Definition 3.1.18 that $\mathsf{Anc}_i(C_i) = \mathsf{Anc}_{i+1}(C_{i+1})\backslash\{\ell\}$. So the lemma also holds in this case. □

Corollary 3.1.23. *Let C_k be a bad cluster at time k. For $1 \leq i \leq k-1$ we define $C_i = \mathsf{prev}(C_{i+1})$. Let $\mathsf{Anc}_k(C_k) = \{\ell_1, \ldots, \ell_s\}$ such that $\ell_{t-1} < \ell_t$ for all $2 \leq t \leq s$ and let $\ell_0 = 0$. Then for any $1 \leq t \leq s$ and for any i with $\ell_{t-1} < i \leq \ell_t$, we have $\{\ell_1, \ldots, \ell_t\} \subset \mathsf{Anc}_i(C_i)$.*

Proof. We prove this via induction over i, starting from $i = k$ in decreasing order. There is nothing to show for $i = k$. For $i < k$ we distinguish two cases. If $\mathsf{Anc}(C_i) = \mathsf{Anc}_{i+1}(C_{i+1})$, the lemma follows from the induction hypothesis.

Otherwise remember that $\mathsf{Anc}_i(C_i) \subset \mathsf{Anc}_k(C_k)$ and $\ell_{t-1} < i$. Thus we know that $\max \mathsf{Anc}_i(C_i) \in \{\ell_t, \ldots, \ell_s\}$ and therefore $\ell_t \leq \max \mathsf{Anc}_i(C_i)$. By Lemma 3.1.22 we know that $\mathsf{Anc}_i(C_i) = \mathsf{Anc}_{i+1}(C_{i+1})\backslash\{\ell\}$, where $\ell = \max \mathsf{Anc}_{i+1}(C_{i+1})$. Thus $\ell_t \leq \max \mathsf{Anc}_i(C_i) < \max \mathsf{Anc}_{i+1}(C_{i+1}) = \ell$ and by induction hypothesis we obtain

$$\{\ell_1, \ldots, \ell_t\} \subset \mathsf{Anc}_{i+1}(C_{i+1})\backslash\{\ell\} = \mathsf{Anc}_i(C_i).$$ □

Before we are able to prove the theorem we need some final lemma.

Lemma 3.1.24. *For every $\epsilon > 0$ there exists $k \in \mathbb{N}$ such that for every $s \in \mathbb{N}$ any sequence of $s+1$ numbers $(\ell_0, \ldots, \ell_s) \in \mathbb{R}^{s+1}_{\geq 0}$ with $\ell_0 = 0$ and $\ell_s = k$ satisfies the following.*

1. *There exists $1 \leq t \leq s$ such that for $\alpha_1 = 4 - \epsilon$ and $\Delta_1 = 1$ we have*

$$\frac{\ell_t + \Delta_1 \sum_{i=0}^{t-1} \ell_i}{\ell_{t-1} + 1} > \alpha_1.$$

2. *There exists $1 \leq t \leq s$ such that for $\alpha_2 = 3 + 2\sqrt{2} - \epsilon$ and $\Delta_2 = 2$ we have*

$$\frac{\ell_t + \Delta_2 \sum_{i=0}^{t-1} \ell_i}{\ell_{t-1} + 1} > \alpha_2.$$

Proof. Let $k, s \in \mathbb{N}$ and $j \in \{1, 2\}$. We call a sequence $(a_0, \ldots, a_s) \in \mathbb{R}^{s+1}_{\geq 0}$ *feasible* if $a_0 = 0, a_s = k$ and for all $1 \leq t \leq s$ we have

$$\frac{a_t + \Delta_j \sum_{i=0}^{t-1} a_i}{a_{t-1} + 1} \leq \alpha_j. \tag{3.1}$$

Our proof is divided in two parts. In the first part we argue that for all $k, s \in \mathbb{N}$ the existence of a feasible sequence $(\ell_0, \ldots, \ell_s)$ yields the existence of a feasible sequence $(b_0, \ldots, b_s)$ which satisfies (3.1) for all $u+1 \leq t \leq s$ with equality, where u is the smallest number such that $b_u \neq 0$. In the second part we observe that there exists $k \in \mathbb{N}$ such that for all $s \in \mathbb{N}$ there is no feasible sequence $(a_0, \ldots, a_s) \in \mathbb{R}^{s+1}_{\geq 0}$ which satisfies (3.1) for all $u+1 \leq t \leq s$ with equality, where u is the smallest number such that $a_u \neq 0$. In combination both parts yield the lemma.

Part 1: Let $k, s \in \mathbb{N}$ and suppose that there exists a feasible sequence $(\ell_0, \dots, \ell_s)$. We consider the set

$$M = \{(a_0, \dots, a_s) \in \mathbb{R}^{s+1}_{\geq 0} \mid (a_0, \dots, a_s) \text{ is feasible}\}$$

of all feasible sequences.

For $(a_0, \dots, a_s) \in M$, we claim that $a_t \leq (\alpha_j + 1)^{t+1}$ for all $0 \leq t \leq s$. We show this via a simple induction over t. If $t = 0$ there is nothing to show since $a_0 = 0$. For $t > 0$ we obtain

$$a_t \leq \alpha_j(a_{t-1} + 1) - \Delta_j \sum_{i=0}^{t-1} a_i \leq \alpha_j(a_{t-1} + 1) \leq \alpha_j((\alpha_j + 1)^t + 1) \leq (\alpha_j + 1)^{t+1}.$$

Here the first inequality follows from the feasibility of the sequence. As a consequence we see that M is a bounded set. Furthermore M is also closed since $a_0 = 0, a_t = k$ are both linear inequalities and (3.1) is a linear inequality for all $1 \leq t \leq s$. Thus M is compact.

We consider the function $F\colon M \to \mathbb{R}$ with $F(a_0, \dots, a_s) = \sum_{i=0}^{s} a_i$. Since F is continuous and M is compact and non-empty we know that F attains a minimum on M, i.e., there is $(b_0, \dots, b_s) \in M$ with $F(b_0, \dots, b_s) \leq F(a_0, \dots, a_s)$ for all $(a_0, \dots, a_s) \in M$. We claim that $(b_0, \dots, b_s)$ satisfies (3.1) with equality for all $u + 1 \leq t \leq s$, where u is the smallest number such that $b_u \neq 0$. Suppose this is not the case and let $u + 1 \leq t \leq s$ be a number such that

$$\frac{b_t + \Delta_j \sum_{i=0}^{t-1} b_i}{b_{t-1} + 1} < \alpha_j.$$

If $b_{t-1} = 0$, then $(0, \dots, 0, b_t, \dots, b_s)$ is also feasible and moreover

$$F(0, \dots, 0, b_t, \dots, b_s) = \sum_{i=t}^{s} b_i < b_u + \sum_{i=t}^{s} b_i \leq F(b_0, \dots, b_s)$$

in contradiction to $(b_0, \dots, b_s)$ being a minimum. Thus we must have $b_{t-1} > 0$ and therefore by continuity there exists an $\epsilon \in (0, b_{t-1})$, such that

$$\frac{b_t + \Delta_j(b_{t-1} - \epsilon) + \Delta_j \sum_{i=0}^{t-2} b_i}{b_{t-1} - \epsilon + 1} \leq \alpha_j.$$

Observe that the sequence $(c_0, \dots, c_s) = (b_0, \dots, b_{t-2}, b_{t-1} - \epsilon, b_t, \dots, b_s)$ is still feasible. The t-th inequality is satisfied by choice of ϵ. All other inequalities are satisfied, since for all $1 \leq r \leq s$ with $r \neq t$ we have

$$\frac{c_r + \Delta_j \sum_{i=0}^{r-1} c_i}{c_{r-1} + 1} \leq \frac{b_r + \Delta_j \sum_{i=0}^{r-1} b_i}{b_{r-1} + 1} \leq \alpha_j.$$

On the other hand

$$F(c_0, \dots, c_s) = \sum_{i=0}^{s} c_i = -\epsilon + \sum_{i=0}^{s} b_i < F(b_0, \dots, b_s),$$

which again stands in contradiction to $(b_0, \dots, b_s)$ being the minimum. Thus $(b_0, \dots, b_s)$ is of the desired form.

Part 2: Let $k, s \in \mathbb{N}$ and $(a_0, \ldots, a_s) \in \mathbb{R}_{\geq 0}^{s+1}$ be a feasible sequence which satisfies (3.1) for all $u+1 \leq t \leq s$ with equality, where u is the smallest number such that $a_u \neq 0$. Thus we know that $a_1 = \ldots = a_{u-1} = 0$ and $a_u \in (0, \alpha_j]$. Furthermore

$$a_{u+1} = \alpha_j(a_u + 1) - \Delta_j \sum_{i=0}^{u} a_i = \alpha_j(a_u + 1) - \Delta_j a_u$$

and for $u+2 \leq t \leq s$ we have

$$\begin{aligned} a_t &= \alpha_j(a_{t-1} + 1) - \Delta_j \sum_{i=0}^{t-1} a_i \\ &= \alpha_j(a_{t-1} + 1) - \Delta_j a_{t-1} - \Delta_j \sum_{i=0}^{t-2} a_i \\ &= \alpha_j(a_{t-1} + 1) - \Delta_j a_{t-1} - (\alpha_j(a_{t-2} + 1) - a_{t-1}) \\ &= \alpha_j(a_{t-1} - a_{t-2}) - (\Delta_j - 1)a_{t-1}. \end{aligned}$$

Here we use that (3.1) is satisfied with equality for t and $t-1$.

Let

$$\Psi = \frac{\alpha_j - \Delta_j + 1 + \sqrt{(\alpha_j - \Delta_j + 1)^2 - 4\alpha_j}}{2}$$

and

$$\Theta = \frac{\alpha_j - \Delta_j + 1 - \sqrt{(\alpha_j - \Delta_j + 1)^2 - 4\alpha_j}}{2}$$

be the two roots of the polynomial $X^2 - (\alpha_j - \Delta_j + 1)X + \alpha_j$. We observe later that $\Phi \neq \Theta$. Let $x = \frac{\Theta a_u - a_{u+1}}{\Theta - \Phi}$ and $y = \frac{a_{u+1} - \Phi a_u}{\Theta - \Phi}$.

Claim: It holds that $a_t = \Phi^{t-u}x + \Theta^{t-u}y$ for all $u \leq t \leq s$.

We prove this claim by induction over t. For $t = u$ we obtain

$$x + y = \frac{\Theta a_u - a_{u+1} + a_{u+1} - \Phi a_u}{\Theta - \Phi} = a_u.$$

For $t = u+1$ we obtain

$$\Phi x + \Theta y = \frac{\Phi\Theta a_u - \Phi a_{u+1} + \Theta a_{u+1} - \Theta\Phi a_u}{\Theta - \Phi} = a_{u+1}.$$

For $t > u+1$ we obtain

$$\begin{aligned} &\Phi^{t-u}x + \Theta^{t-u}y \\ &= \Phi^{t-u-2}x((\alpha_j - \Delta_j + 1)\Phi - \alpha_j) + \Theta^{t-u-2}y((\alpha_j - \Delta_j + 1)\Theta - \alpha_j) \\ &= \alpha_j((\Phi^{t-u-1}x + \Theta^{t-u-1}y) - (\Phi^{t-u-2}x + \Theta^{t-u-2}y)) - (\Delta_j - 1)(\Phi^{t-u-1}x + \Theta^{t-u-1}y) \\ &= \alpha_j(a_{t-1} - a_{t-2}) - (\Delta_j - 1)a_{t-1} \\ &= a_t. \end{aligned}$$

For the first equality we used that Φ and Θ are roots of $X^2 - (\alpha_j - \Delta_j + 1)X + \alpha_j$, i.e., $\Phi^2 = (\alpha_j - \Delta_j + 1)\Phi - \alpha_j$ and $\Theta^2 = (\alpha_j - \Delta_j + 1)\Theta - \alpha_j$. For the third equality we used the induction hypothesis. This proves the claim.

We argue that if k is large enough, there must be $u \leq t \leq s$ with $a_t < 0$ in contradiction to our assumption that $(a_0, \ldots, a_s)$ is feasible. For this we observe that by

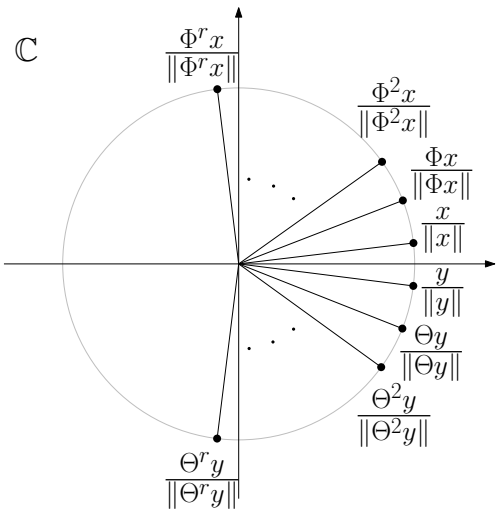

Figure 3.7: Here we see the normalized numbers on the complex plane.

choice of α_j and Δ_j, we get $(\alpha_j - \Delta_j + 1)^2 - 4\alpha_j < 0$ and thus Φ and Θ are complex numbers. Furthermore Φ and Θ are complex conjugates and so are x and y. Thus there exists $r > 0$ such that the real part of $\Phi^r x$ and $\Theta^r y$ is negative and thus $\Phi^r x + \Theta^r y$ is negative, see Figure 3.7.

Observe that $a_t \leq (\alpha_j + 1)^{t-u+1}$ for $u \leq t \leq s$. One can prove this similar to the bound in Part 1. Thus if $k \geq (\alpha_j + 1)^{r+1}$ we obtain $s \geq r + u$ and $a_{r+u} = \Phi^r x + \Theta^r y$ is negative. Therefore $(a_0, \ldots, a_s)$ is not feasible in contradiction to our assumption.

Let now $k \geq (\alpha_j + 1)^{r+1}$ and suppose there exists $s \in \mathbb{N}$ and a feasible sequence $(\ell_0, \ldots, \ell_s)$. By the first part we know that there also exists a feasible sequence $(a_0, \ldots, a_s)$ which satisfies (3.1) for all $u + 1 \leq t \leq s$ with equality, where u is the smallest number such that $a_u \neq 0$. This is in contradiction with the second part, where we prove that for $k \geq (\alpha_j + 1)^{r+1}$ such a sequence cannot exist. □

Theorem 3.1.8. *For* $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}\}$ *we have* $\rho_{\mathsf{cost}} \geq 3 + 2\sqrt{2}$ *and for* $\mathsf{cost} = \mathsf{drad}$ *we have* $\rho_{\mathsf{cost}} \geq 4$.

Proof. Let $\epsilon > 0$ and k be the respective number from Lemma 3.1.24. We claim that the approximation factor of any hierarchical clustering $\mathscr{H} = (\mathcal{H}_{N_k}, \ldots, \mathcal{H}_1)$ on the instance $(\mathcal{X}, \mathcal{P}, d)$ is larger than $3 + 2\sqrt{2} - \epsilon$ if $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{rad}\}$ and larger than $4 - \epsilon$ if $\mathsf{cost} = \mathsf{drad}$. First we use Lemma 3.1.16 to observe that there is a cluster $C_k \in \mathcal{H}_{\frac{N_k}{\Gamma N_{k-1}}}$ that is bad at time k. For $1 \leq i \leq k-1$ we define $C_i = \mathsf{prev}(C_{i+1})$. Let $\mathsf{Anc}_k(C_k) = \{\ell_1, \ldots, \ell_s\}$ with $\ell_{t-1} < \ell_t$ for $2 \leq t \leq s$ and let $\ell_0 = 0$. We know by Corollary 3.1.23, that for any $1 \leq t \leq s$ and for $i = \ell_{t-1} + 1$ we have $\{\ell_1, \ldots, \ell_t\} \subset \mathsf{Anc}_i(C_i)$. Let $\ell' = \max \mathsf{Anc}_i(C_i)$,

we obtain by Lemma 3.1.21 and Lemma 3.1.20 that

$$\begin{aligned}
\mathsf{diam}(C_i) &\geq \ell' + 2 \sum_{a \in \mathsf{Anc}_i(C_i) \setminus \{\ell'\}} a \geq \ell_t + 2 \sum_{u=1}^{t-1} \ell_u, \\
\mathsf{rad}(C_i) &\geq \frac{\mathsf{diam}(C_i)}{2} \geq \frac{\ell_t + 2\sum_{u=1}^{t-1} \ell_u}{2}, \\
\mathsf{drad}(C_i) &\geq \sum_{a \in \mathsf{Anc}_i(C_i)} a \geq \sum_{u=1}^{t} \ell_u.
\end{aligned}$$

Remember that by Corollary 3.1.13 $\mathcal{A}_i$ is an optimal $\frac{N_k}{\Gamma N_{i-1}}$-clustering with $\mathsf{cost}(\mathcal{A}_i) = i$ if $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{drad}\}$ and $\mathsf{cost}(\mathcal{A}_i) = i/2$ if $\mathsf{cost} = \mathsf{rad}$. We obtain

$$\begin{aligned}
\frac{\mathsf{rad}(C_i)}{\mathsf{rad}(\mathcal{A}_i)} = \frac{2\mathsf{rad}(C_i)}{2\mathsf{rad}(\mathcal{A}_i)} \geq \frac{\mathsf{diam}(C_i)}{\mathsf{diam}(\mathcal{A}_i)} \geq \frac{\ell_t + 2\sum_{u=1}^{t-1} \ell_u}{\ell_{t-1} + 1} \\
\frac{\mathsf{drad}(C_i)}{\mathsf{drad}(\mathcal{A}_i)} \geq \frac{\sum_{u=1}^{t} \ell_u}{\ell_{t-1} + 1}
\end{aligned}$$

which are lower bounds on the approximation factor of $\mathscr{H}$.

We apply Lemma 3.1.24 on $(\ell_0, \ldots, \ell_s)$ to observe that there is $1 \leq t' \leq s$ such that

$$\frac{\ell_{t'} + 2\sum_{u=1}^{t'-1} \ell_u}{\ell_{t'-1} + 1} > 3 + 2\sqrt{2} - \epsilon$$

and an $1 \leq t'' \leq s$ such that

$$\frac{\sum_{u=1}^{t''} \ell_u}{\ell_{t''-1} + 1} > 4 - \epsilon.$$

This proves the theorem. □

3.1.3 Counterexample for Mondal's Algorithm

So far we learned that the price of hierarchy is $3 + 2\sqrt{2}$ for $\mathsf{rad}, \mathsf{diam}$ and 4 for drad. However, the price of hierarchy does not reveal whether one can compute hierarchical clusterings with these approximation guarantees. The best polynomial time algorithm is the 8-approximation by Dasgupta and Long [39]. Mondal claims that one can even obtain a 6-approximation for the discrete radius objective [66, Theorem 3.7]. We claim that this is not correct and present an example where the approximation factor is 7. First we give a brief summary of Mondal's algorithm.

Let $(\mathcal{X}, \mathcal{P}, d)$ be the clustering instance. In the beginning we compute a numbering of the points in $\mathcal{P}$ by running Gonzales' algorithm [43]. The numbering is computed as follows. We pick the first point $x_1 \in \mathcal{P}$ arbitrarily and set $R_1 = \infty$. For $2 \leq k \leq |\mathcal{P}|$ we set

$$x_k = \mathrm{argmax}_{x \in \mathcal{P} \setminus \{x_1, \ldots, x_{k-1}\}} \min_{1 \leq i \leq k-1} d(x, x_i)$$

and $R_k = \min_{1 \leq i \leq k-1} d(x_k, x_i)$. In other words the k-th point is picked as far as possible from the points $x_1, \ldots, x_{k-1}$ and we denote by R_k the distance of x_k to $x_1, \ldots, x_{k-1}$.

Based on the R-values we define the parent of a point $x \in \mathcal{P} \setminus \{x_1\}$. Let $N(x) = \mathrm{argmin}\{d(x, y) \mid y \in \mathcal{P}, R_x \leq \frac{R_y}{2}\}$ denote the parent of x. In other words $N(x)$ is the

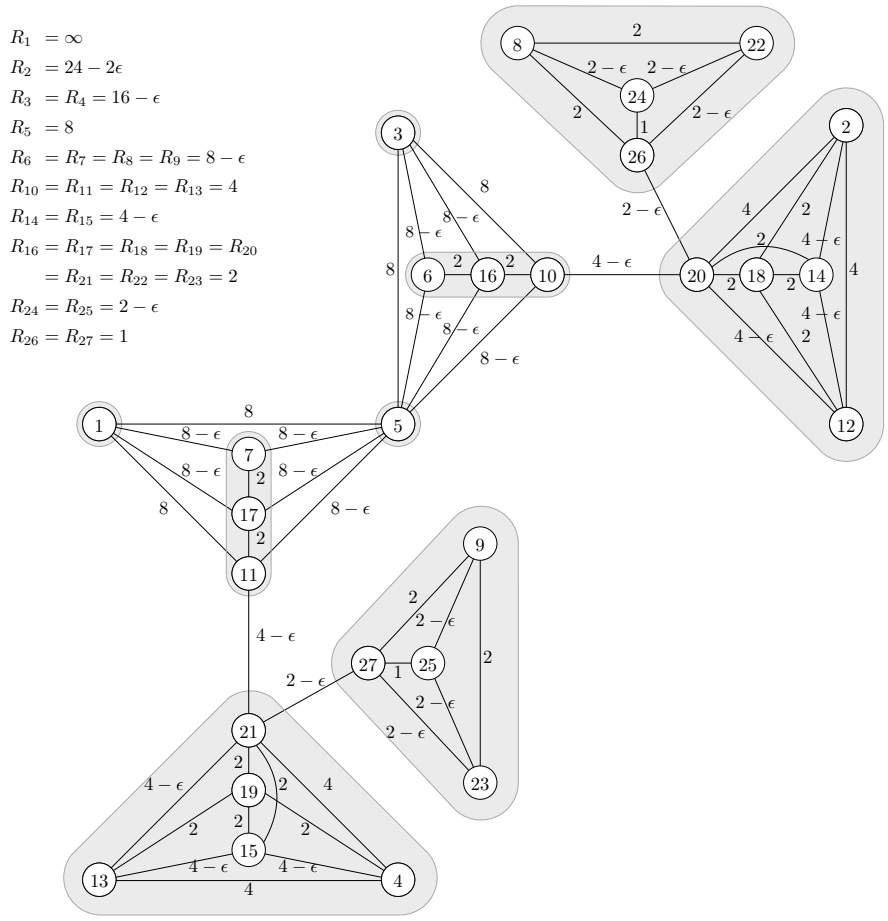

Figure 3.8: Here we see the clustering instance and the numbering obtained from Gonzales' algorithm as well as the optimal 9-clustering with radius 2 depicted in gray.

point nearest to x that satisfies $R_x \leq \frac{R_{N(x)}}{2}$. Notice that every point in $\mathcal{P}\backslash\{x_1\}$ has a properly defined parent, as $R_1 = \infty$.

We build a tree on $\mathcal{P}$ as follows. For every point $x \in \mathcal{P}$ we simply add an edge between x and $N(x)$. The resulting graph is cycle free, since $R_x < R_{N(x)}$ for all $x \in \mathcal{P}$, and contains $|\mathcal{P}| - 1$ edges. Thus it is indeed a tree.

For any given $1 \leq k \leq |\mathcal{P}|$ we observe that by deleting the edges $\{x_i, N(x_i)\}$ for all $2 \leq i \leq k$ the tree decomposes into k connected components with vertex sets $H_k^1, \ldots, H_k^k$. We define the k-clustering on $\mathcal{P}$ to be $\mathcal{H}_k = (H_k^1, \ldots, H_k^k)$. Then $\mathscr{H} = (\mathcal{H}_{|\mathcal{P}|}, \ldots, \mathcal{H}_1)$ is a hierarchical clustering of $\mathcal{P}$.

We believe that the algorithm by Mondal does not differ significantly from the algorithm by Dasgupta and Long. Since we already know that the analysis of the approximation guarantee of Dasgupta and Long's algorithm is tight [37], the significant improvement on the approximation guarantee seems surprising. We present an example where Mondal's algorithm in fact computes a $7 - \epsilon$ approximation for some arbitrarily small $\epsilon > 0$, contradicting the claimed approximation guarantee of 6. We believe that this example can be generalized to prove that the approximation guarantee of Mondal's algorithm is at least 8.

Let $\epsilon \in (0, \frac{1}{2})$. Figure 3.8 shows a graph with 27 points that need to be clustered.

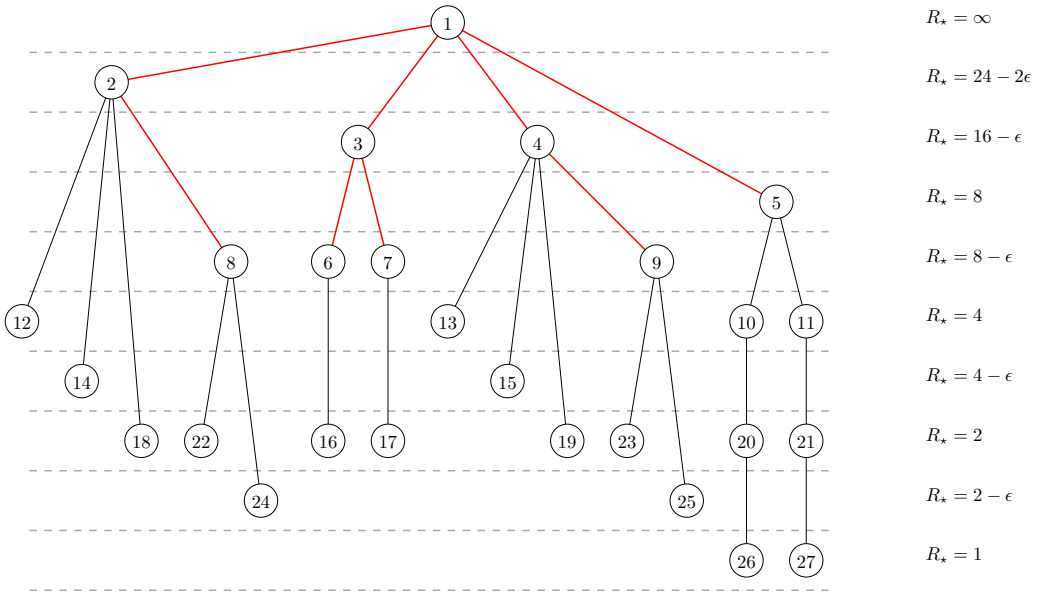

Figure 3.9: Here we see the final tree. To obtain the 9-clustering we cut the red edges. The resulting clustering contains the cluster $\{x_5, x_{10}, x_{11}, x_{20}, x_{21}, x_{26}, x_{27}\}$ of radius $14 - 3\epsilon$.

The metric is given by the shortest-path metric in the graph. We perform Mondal's algorithm on this instance under the assumption that we can decide how to break ties, whenever they occur.

In Figure 3.8 we see the numbering of the points which is computed by Gonzales' algorithm as well as all R-values. Figure 3.9 shows the resulting tree. We obtain the 9-clustering by cutting all edges $\{x_i, N(x_i)\}$ with $2 \leq i \leq 9$. This clustering contains the cluster $\{x_5, x_{10}, x_{11}, x_{20}, x_{21}, x_{26}, x_{27}\}$, whose radius is $14 - 3\epsilon$, while the radius of the optimal 9-clustering is 2 (see Figure 3.8).

3.2 Complete Linkage in General Metric Spaces

Complete linkage is a popular algorithm for computing hierarchical clusterings. Given a set of n points $\mathcal{P}$ contained in a metric space $(\mathcal{X}, d)$, the algorithm starts with the n-clustering $\mathcal{H}_n$ where every point is contained in its own cluster. In every step complete linkage then merges the two clusters whose merge results in the smallest increase of the objective function $\mathsf{cost} \in \{\mathsf{drad}, \mathsf{rad}, \mathsf{diam}\}$, i.e., given a $(k+1)$-clustering $\mathcal{H}_{k+1}$ complete linkage merges the two clusters $A, B \in \mathcal{H}_{k+1}$ that minimize $\mathsf{cost}(A \cup B)$. This process terminates when only one cluster is left.

The single linkage algorithm is also a popular agglomerative clustering method. Instead of minimizing the given objective function single linkage merges the two clusters with smallest distance to each other. Given a $(k+1)$-clustering $\mathcal{H}_{k+1}$ single linkage merges the two clusters $A, B \in \mathcal{H}_{k+1}$ that minimize $d(A, B) = \min_{x \in A, y \in B} d(x, y)$. In this section we analyze the approximation guarantee for complete linkage and single linkage for the objectives drad and diam.

One of the biggest and most well-known issues concerning single linkage is that of chaining. If there is a sequence of points $x_1, \ldots, x_k \in \mathcal{P}$ with $d(x_i, x_{i+1})$ relatively small for all i, then single linkage might merge all of them together, despite the resulting cluster being quite large. Dasgupta and Long [39] show with their lower bound of $\Omega(\log(k))$ that a similar process of chaining can also occur when executing complete linkage. They give the example of points placed on a regular $(k \times k)$-grid with a spacing of 1. The distance is given by the sum of the discrete metric on the horizontal axis and the logarithm of the absolute value of the vertical axis. That is, $d((x, y), (x', y')) = \mathbf{1}_{x \neq x'} + \log_2(1 + |y - y'|)$. Now, although an optimal clustering just consists of the individual rows of the grid, complete linkage might reproduce the columns instead (assuming that k is a power of 2): iteratively go from top to bottom and merge vertically neighboring clusters. Every such iteration halves the number of clusters and, due to the logarithm, only increases the cost by 1, just as when merging along the rows. Of course, we would have to pay only once to merge horizontally, whereas we have to pay $\log_2(k)$ times to merge vertically, but complete linkage cannot distinguish between these two cases. In fact, one can shift the vertical placement by arbitrarily small values to ensure that complete linkage always chooses the bad case.

We have to heavily modify the example to improve upon this $\log_2(k)$ factor. The fundamental problem is this: a vertical merge is only allowed to increase the cost by 1 to tie it with any horizontal merge, whereas the number of rows occupied by a cluster (and thus its diameter) doubles. We raise the lower bound by constructing an instance on which complete linkage iteratively merges diagonally shifted clusters. This process of merging clusters is much slower and does not require us to introduce a logarithmic scaling: merging one such cluster into the other incurs a cost of 1, while at the same time increasing the number of occupied rows only by one. The instance that we describe later is successively built from smaller components that exhibit exactly this behaviour, while ensuring that any such merge does not pay for the whole row.

Following the work [1] one can show for complete linkage an upper bound of $\log_2(|P| - k)$ for drad. This comes from the following easy property, which is true for the radius but cannot be transferred to diameter: Suppose the optimal drad solution $\mathcal{O}$ has cost x. In a complete linkage clustering consisting of more than k clusters two of its centers must lie in the same optimal cluster and therefore are at distance $\leq 2x$ to each other. Thus the merge that is performed by complete linkage increases the cost by at most $2x$. However

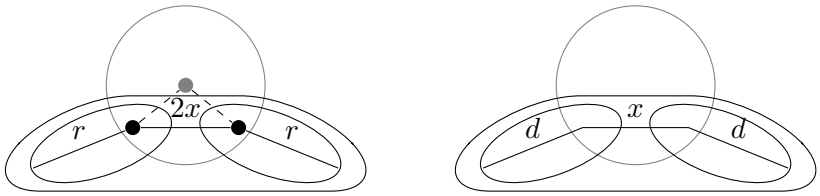

Figure 3.10: On the left we see two clusters with radius r whose centers lie in the same optimal cluster with radius x. The radius of the merged cluster is at most $r + 2x$. On the right we have a similar situation for the diameter but we can upper bound the diameter of the merged cluster only by $2d + x$.

if we replace drad by diam we see that the cost is more than doubled in the worst case (see Figure 3.10), which is not enough to obtain an upper bound polynomial in k. Thus we introduce another perspective on the cost of a cluster. A cluster is good if its cost is small enough in comparison to the number of optimal clusters from $\mathcal{O}$ which it intersects. As $\mathcal{O}$ consists of k clusters this already implies a sufficiently small upper bound for good clusters. For all remaining clusters we show that their number is small enough. This approach leads to an upper bound of $O(k^{\ln(3)/\ln(2)})$ for diam and, in combination with the $\log_2(|P| - k)$ upper bound, an upper bound of $O(k)$ for drad.

To analyze the hierarchical clustering $\mathscr{H}$ constructed by complete linkage we often consider the smallest clustering from $\mathscr{H}$ (in terms of the number of clusters) whose cost does not exceed a given bound. This perspective is already used in [44] and allows for a better handling of the cost.

Definition 3.2.1. *Let* $\mathsf{cost} \in \{\mathsf{drad}, \mathsf{diam}\}$ *and let* $\mathscr{H} = (\mathcal{H}_n, \dots, \mathcal{H}_1)$ *be the hierarchical clustering computed by complete linkage with respect to* $\mathsf{cost} \in \{\mathsf{drad}, \mathsf{diam}\}$*. For any* $x \geq 0$ *let* $t_{\leq x} = \min\{k \mid \mathsf{cost}(\mathcal{H}_k) \leq x\}$ *and set* $\mathcal{C}_x = \mathcal{H}_{t_{\leq x}}$*. Furthermore, for a cluster* $C \subseteq \mathcal{P}$ *and an optimal* k*-clustering* $\mathcal{O} = \mathcal{O}_k$ *we denote by* $\mathcal{O}_C = \{O \in \mathcal{O} \mid O \cap C \neq \emptyset\}$ *the set of all optimal* k*-clusters hit by* C*.*

Observe that $\mathcal{C}_x$ is the smallest clustering from $\mathscr{H}$ with cost at most x. Thus it has the useful property that every merge of two clusters in $\mathcal{C}_x$ results in a clustering of cost more than x.

3.2.1 Approximation Guarantee of Single Linkage

As outlined in [39] there are clustering instances where single linkage builds chains yielding the lower bound $\Omega(k)$ on the approximation factor. We show that this is the worst case scenario, as in fact single linkage computes an $O(k)$-approximation for diam and drad.

Let $(\mathcal{H}_k)_{k=1}^n$ be the hierarchical clustering computed by single linkage on $(\mathcal{P}, d)$. Recall that $\mathcal{H}_{k-1}$ arises from $\mathcal{H}_k$ by merging two clusters $A, B \in \mathcal{H}_k$ that minimize $d(A, B)$.

We first compare the radius of $\mathcal{H}_k$ to the radius of an optimal k-clustering $\mathcal{O}$ with respect to drad. We introduce a graph G whose vertices are the optimal clusters $V(G) = \mathcal{O}$ and whose edges $E(G) = \{\{O, O'\} \subseteq \mathcal{O} \mid d(O, O') \leq 2\mathsf{drad}(\mathcal{O})\}$ connect all pairs of optimal clusters $O, O' \in \mathcal{O}$ with distance at most twice the optimal radius.

We make a similar construction to compare the diameter of $\mathcal{H}_k$ to the diameter of an optimal diam clustering $\mathcal{O}'$. We consider the graph G' with $V(G') = \mathcal{O}'$ where two clusters in $\mathcal{O}'$ are connected via an edge if their distance is at most $\mathsf{diam}(\mathcal{O}')$.

To estimate the cost of a single linkage cluster $C \in \mathcal{H}_k$ we look at the optimal clusters hit by C. The next lemma shows that for any two points in C we can find a path connecting them that passes through a chain of optimal clusters with distance at most $2\mathsf{drad}(\mathcal{O})$ or $\mathsf{diam}(\mathcal{O}')$ when considering the radius or diameter, respectively. One can already anticipate that this gives an upper bound of $O(k)\mathsf{drad}(\mathcal{O})$ or $O(k)\mathsf{diam}(\mathcal{O}')$ on the radius or diameter of any such cluster C. In Figure 3.11 we see an example of such a cluster C and the optimal clusters hit by C.

Lemma 3.2.2. *Let $C \in \mathcal{H}_t$ be a cluster computed by single linkage at a time step $t \geq k$. Then the graphs $G[\mathcal{O}_C]$ and $G'[\mathcal{O}'_C]$ induced by the vertex set of optimal clusters hit by C are connected.*

Proof. We prove the lemma for $G[\mathcal{O}_C]$ by induction. At the beginning ($t = n$) the lemma obviously holds, since any cluster contained in $\mathcal{H}_n$ is a point and thus hits only one single optimal cluster. Assume now that the claim holds for $t > k$. By the pigeonhole principle there must exist two clusters $C, C' \in \mathcal{H}_t$ with two points $c \in C$ and $c' \in C'$ lying in the same optimal cluster $O \in \mathcal{O}$. We know that $d(C, C') \leq 2\mathsf{drad}(O) \leq 2\mathsf{drad}(\mathcal{O})$. But this value is exactly the objective that single linkage minimizes, so we know in particular that this upper bound also holds for the distance between the clusters D, D' chosen by single linkage. Combining this with the induction hypothesis that both $G[\mathcal{O}_D]$ and $G[\mathcal{O}_{D'}]$ are connected finishes the proof. One proves analogously that $G'[\mathcal{O}'_C]$ is connected by replacing $d(C, C') \leq 2\mathsf{drad}(O) \leq 2\mathsf{drad}(\mathcal{O})$ through $d(C, C') \leq \mathsf{diam}(\mathcal{O}')$ in the above argument. □

As we see in Figure 3.11 this already yields an upper bound of $2k\mathsf{diam}(\mathcal{O}')$ on the diameter of C. We estimate the radius of C by looking at the paths going through optimal clusters in $\mathcal{O}_C$ that are at distance at most $2\mathsf{drad}(\mathcal{O})$ from one another. Choosing the center appropriately and uncoiling these paths in our original space $\mathcal{P}$ yields our upper bound of $(2k + 2)\mathsf{drad}(\mathcal{O})$.

Theorem 3.2.3. *Let $\mathsf{cost} \in \{\mathsf{drad}, \mathsf{diam}\}$ and $(\mathcal{H}_k)_{k=1}^n$ be the hierarchical clustering computed by single linkage on $(\mathcal{P}, d)$ and let $\mathcal{O}_k$ be an optimal clustering with respect to cost. We have for all $1 \leq k \leq n$*

1. $\mathsf{cost}(\mathcal{H}_k) \leq (2k + 2) \cdot \mathsf{cost}(\mathcal{O}_k)$ *if* $\mathsf{cost} = \mathsf{drad}$
2. $\mathsf{cost}(\mathcal{H}_k) \leq 2k \cdot \mathsf{cost}(\mathcal{O}_k)$ *if* $\mathsf{cost} = \mathsf{diam}$.

Proof. We prove the statement for drad. Fix an arbitrary time step $1 \leq k \leq n$ and denote $\mathcal{O} = \mathcal{O}_k$. Let $C \in \mathcal{H}_k$ be an arbitrary cluster and Q a longest simple path in $G[\mathcal{O}_C]$. Choose as center for C an arbitrary point $c \in C \cap O$ from an optimal cluster O lying in the middle of Q. Note that by this choice every other vertex in $G[\mathcal{O}_C]$ is reachable from O by a path of length at most $\frac{k}{2}$. Uncoiling such paths in $\mathcal{P}$ gives us an upper bound of $2(k + 1)\mathsf{drad}(\mathcal{O})$ for the distance between c and any other point $z \in C$ as follows: If $O_z \in \mathcal{O}$ is the optimal cluster containing z, then by choice of O, there exists a path $O = O_1, \ldots, O_{\ell+1} = O_z$ in $G[\mathcal{O}_C]$ of length $\ell \leq \frac{k}{2}$ connecting them.

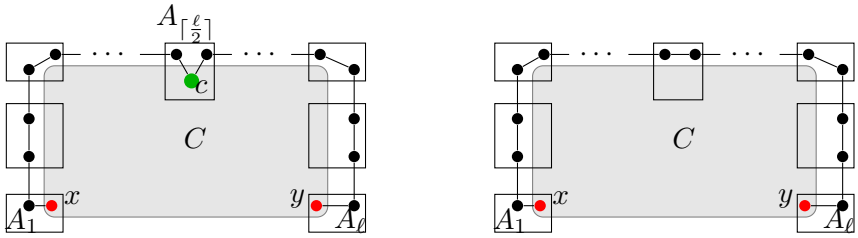

Figure 3.11: Cluster C hits the optimal clusters $A_1, \dots, A_\ell$ with $d(A_i, A_{i+1}) \leq 2\mathsf{drad}(\mathcal{O})$ when considering the radius on the left and $d(A_i, A_{i+1}) \leq \mathsf{diam}(\mathcal{O}')$ when considering the diameter on the right. In the left picture, we see that choosing center c in $A_{\lceil \frac{\ell}{2} \rceil}$ leads to $\mathsf{drad}(C) \leq 2(\ell+1)\mathsf{drad}(\mathcal{O})$. Similarly the diameter of C in the right picture is at most $2\ell\mathsf{drad}(\mathcal{O})$.

That means, for each $i = 1, \dots, \ell$ there exist points $x_i \in O_i, y_{i+1} \in O_{i+1}$ such that $d(x_i, y_{i+1}) \leq 2\mathsf{drad}(\mathcal{O})$. Hence

$$\begin{aligned} d(c,z) &\leq d(c,x_1) + \sum_{i=1}^{\ell-1} \left(d(x_i, y_{i+1}) + d(y_{i+1}, x_{i+1})\right) \\ &\quad + d(x_\ell, y_{\ell+1}) + d(y_{\ell+1}, z) \\ &\leq 2(2\ell+1)\mathsf{drad}(\mathcal{O}) \leq 2(k+1)\mathsf{drad}(\mathcal{O}). \end{aligned}$$

Using Lemma 3.2.2 one proves the statement for diam analogously. □

3.2.2 An Upper Bound for Complete Linkage

Even though complete linkage is often used when it comes to computing a hierarchical clustering, there are no known non-trivial upper bounds for its approximation guarantee in general metric spaces, to the best of the author's knowledge. We give an upper bound for complete linkage for the radius and diameter objective.

An Upper Bound for Radius-Based Cost

We show that the approximation ratio of complete linkage for the drad objective is in $O(k)$.

Theorem 3.2.4. *Let $\mathscr{H} = (\mathcal{H}_n, \dots, \mathcal{H}_1)$ be the hierarchical clustering computed by complete linkage on $(\mathcal{P}, d)$ with respect to drad. For all $1 \leq k \leq n$ the radius $\mathsf{drad}(\mathcal{H}_k)$ is upper bounded by $O(k)\mathsf{drad}(\mathcal{O}_k)$, where $\mathcal{O}_k$ is an optimal k-clustering with respect to drad.*

To simplify the notation we fix an arbitrary k and assume that the optimal k-clustering $\mathcal{O} = \mathcal{O}_k$ has radius $\mathsf{drad}(\mathcal{O}) = \frac{1}{2}$. The latter is possible without loss of generality by scaling the metric appropriately.

We split the proof of Theorem 3.2.4 into two parts. In the first, we derive a crude upper bound for the increasing cost of clusterings produced during the execution of complete linkage. This part follows from [1], which uses the same bound to estimate the

cost of some few merge steps. Lemma 3.2.7 shows that the difference in cost between $\mathcal{H}_k$ and $\mathcal{H}_t$ for $t > k$ is at most $\lceil \log_2(t-k) \rceil + 1$. That is, $\mathsf{drad}(\mathcal{H}_k) \leq \lceil \log_2(t-k) \rceil + 1 + \mathsf{drad}(\mathcal{H}_t)$ holds for all $1 \leq k < t \leq n$. A clustering $\mathcal{H}_t$ whose radius we can estimate directly (i.e. without refering to any other clustering) thus yields a proper upper bound for $\mathsf{drad}(\mathcal{H}_k)$. Ideally, this clustering should consist of relatively few clusters (so that $\lceil \log_2(t-k) \rceil$ is small), while at the same time not being too expensive. Of course, however, these criteria oppose each other. Naively choosing the initial clustering $\mathcal{H}_t = \mathcal{H}_n$ is not good enough. Although its radius is minimal, the number of clusters is too high, only yielding an upper bound of $\mathsf{drad}(\mathcal{H}_k) \leq \lceil \log_2(n-k) \rceil + 1$. In the second part of the proof we thus set out to find a different clustering to start from.

Part 1: An estimate of the relative difference in cost When dealing with radii, any merge done by complete linkage previous to reaching a k-clustering increases the radius by at most $2\mathsf{drad}(\mathcal{O}) = 1$ (Figure 3.10). This is due to the fact that the centers of two of those clusters are contained in the same optimal cluster.

We show that complete linkage clusterings at times $t_{\leq x}$ and $t_{\leq x+1}$ can have at most k clusters in common. All other clusters from $\mathcal{C}_x$ are merged in $\mathcal{C}_{x+1}$.

Lemma 3.2.5. *For all $x \geq 0$ the clustering $\mathcal{C}_{x+1}$ contains at most k clusters of radius at most x. In particular, it holds that $|\mathcal{C}_{x+1} \cap \mathcal{C}_x| \leq k$.*

Proof. Assume on the contrary that at time $t_{\leq x+1}$ there exist $k+1$ pairwise different clusters $D_1, \ldots, D_{k+1}$ of radius at most x. Denote by $d_i \in D_i$ a point that induces the smallest radius, i.e. $\mathsf{drad}(D_i) = \max_{d \in D_i} d(d, d_i)$ for all i. Then two of these points, say d_1 and d_2, have to be contained in the same optimal cluster $O \in \mathcal{O}$. Hence, we know that

$$\mathsf{drad}(D_1 \cup D_2) \leq 1 + \max_{i \in \{1,2\}} \mathsf{drad}(D_i) \leq 1 + x$$

because $d(d_1, d_2) \leq 2\mathsf{drad}(O) \leq 2\mathsf{drad}(\mathcal{O}) = 1$ and $\mathsf{drad}(D_i) \leq x$ for $i = 1, 2$. This contradicts the definition of $\mathcal{C}_{x+1}$, as D_1 and D_2 can still be merged without pushing the radius beyond $x + 1$. □

With this we can upper bound $|\mathcal{C}_{x+i}|$ in terms of $|\mathcal{C}_x|$ for all $i \in \mathbb{N}$.

Corollary 3.2.6. *For all $i \in \mathbb{N}$ and $x \geq 0$ it holds that $|\mathcal{C}_{x+i}| \leq k + \frac{1}{2^i}(|\mathcal{C}_x| - k)$.*

Proof. First, we consider what happens when we increase the radius by 1. We fix an arbitrary $x' \geq 0$. Lemma 3.2.5 shows that at most k clusters from $\mathcal{C}_{x'}$ are left untouched, while the remaining $|\mathcal{C}_{x'}| - k$ clusters have to be merged with at least one other cluster (thus at least halving the number of those clusters) to get to $\mathcal{C}_{x'+1}$. This yields a bound of

$$|\mathcal{C}_{x'+1}| \leq k + \frac{1}{2}(|\mathcal{C}_{x'}| - k).$$

Now, the case for general $i \in \mathbb{N}$ follows by a straightforward induction. We have just shown that the claim is true for $i = 1$, where we set $x' = x$. For the induction step suppose that

$$|\mathcal{C}_{x+i-1}| \leq k + \frac{1}{2^{i-1}}(|\mathcal{C}_x| - k).$$

Substituting this into the inequality

$$|\mathcal{C}_{x+i}| \leq k + \frac{1}{2}(|\mathcal{C}_{x+i-1}| - k),$$

derived from the first part of our proof with $x' = x + i - 1$, yields

$$|\mathcal{C}_{x+i}| \leq k + \frac{k + \frac{1}{2^{i-1}}(|\mathcal{C}_x| - k) - k}{2} = k + \frac{1}{2^i}(|\mathcal{C}_x| - k)$$

as claimed. □

Lemma 3.2.7. *For all $k < t \leq n$ it holds that $\mathsf{drad}(\mathcal{H}_k) \leq \lceil \log_2(t-k) \rceil + 1 + \mathsf{drad}(\mathcal{H}_t)$.*

Proof. Let $x = \mathsf{drad}(\mathcal{H}_t)$, so that $\mathcal{C}_x$ consists of at most t clusters. Applying Corollary 3.2.6 with $i = \lceil \log_2(t-k) \rceil + 1$ then shows that

$$|\mathcal{C}_{x+i}| < k + \frac{1}{t-k}(|\mathcal{C}_x| - k) \leq k + 1.$$

That is, $\mathcal{C}_{x+i}$ emerges from $\mathcal{H}_k$ by merging some (or none) of its clusters and we can conclude that $\mathsf{drad}(\mathcal{H}_k) \leq \mathsf{drad}(\mathcal{C}_{x+i}) \leq x + i = \mathsf{drad}(\mathcal{H}_t) + \lceil \log_2(t-k) \rceil + 1$. □

Part 2: A cheap clustering with few clusters Suppose that there exists a complete linkage clustering $\mathcal{H}_t$ for some $t > k$ with $t \in O(2^k)$ clusters and $\mathsf{drad}(\mathcal{H}_t) \in O(k)$. Then applying Lemma 3.2.7 shows that

$$\mathsf{drad}(\mathcal{H}_k) \in \log_2(O(2^k)) + 1 + O(k) = O(k) = O(k)\mathsf{drad}(\mathcal{O})$$

and Theorem 3.2.4 is proven (recall that $\mathsf{drad}(\mathcal{O}) = \frac{1}{2}$). We show that $\mathcal{H}_t = \mathcal{C}_{4k+2}$ is a sufficiently good choice. To estimate the size of $\mathcal{C}_{4k+2}$, we distinguish between active and inactive clusters. Remember that $\mathcal{O}_C = \{O \in \mathcal{O} \mid O \cap C \neq \emptyset\}$ is the set of optimal clusters hit by C.

Definition 3.2.8. *We call a cluster $C \in \mathcal{C}_x$* active, *if $\mathsf{drad}(C) \leq 4 \cdot |\mathcal{O}_C|$, or if there exists a cluster $C' \in \bigcup_{i=1}^{x} \mathcal{C}_{x-i}$ such that $\mathcal{O}_C = \mathcal{O}_{C'}$ and $\mathsf{drad}(C') \leq 4 \cdot |\mathcal{O}_{C'}|$. Otherwise, C is called* inactive.

The behavior that makes complete linkage difficult to analyze it the problem that complete linkage can merge clusters that are quite far apart. That is complete linkage can produce clusters that are very expensive relative to the number of optimal clusters hit by them. We mark such clusters as inactive and count them directly the first time they are created. We will see that the number of such clusters is small. However the number of active clusters is potentially large, but if the radius of the clustering reaches $4k + 2$, this number can also be bounded as we see in the following lemma.

Lemma 3.2.9. *There are at most 2^k active clusters in $\mathcal{C}_{4k+2}$.*

Proof. Notice that at time $t_{\leq 4k+2}$ there cannot exist two active clusters C_1 and C_2 with $\mathcal{O}_{C_1} \subseteq \mathcal{O}_{C_2}$. Indeed, since C_2 hits all the optimal clusters hit by C_1 we get that

$$\mathsf{drad}(C_1 \cup C_2) \leq \mathsf{drad}(C_2) + 1 \leq 4|\mathcal{O}_{C_2}| + 2 \leq 4k + 2$$

where the second inequality follows from the fact that C_2 is active. We conclude that C_1 and C_2 would have been merged in $\mathcal{C}_{4k+2}$. Now, if there are more than 2^k active clusters in $\mathcal{C}_{4k+2}$, then at least two of them must hit exactly the same set of optimal clusters. Since we have just ruled this out, the lemma follows. □

We estimate the number of inactive clusters by looking at the circumstances under which they arise. As it happens, at each step there are not many clusters whose merge yields an inactive cluster.

Lemma 3.2.10. *There are at most $4k^2 + k$ inactive clusters in $\mathcal{C}_{4k+2}$.*

Proof. Let m_x be the number of inactive clusters in $\mathcal{C}_x$. We show that the recurrence relation $m_x \leq m_{x-1} + k$ holds for any $x \in \mathbb{N}$. In that case $m_{4k+2} \leq (4k+1)k = 4k^2 + k$ since $m_1 = 0$ and we are done.

To prove the recurrence relation first fix some arbitrary $x \in \mathbb{N}$ and let $D \in \mathcal{C}_x$ be an inactive cluster. Let $D_1, \ldots, D_\ell \in \mathcal{C}_{x-1}$ be the clusters whose merge results in D. We show that none of them can be active at time $t_{\leq x-1}$ and have radius at least $x-2$. Since this only leaves few possible clusterings, we get the recurrence inequality given above. Suppose that for one of the clusters, say D_i, it holds that $4 \cdot |\mathcal{O}_{D_i}| + 1 \geq \mathsf{drad}(D_i) \geq x - 2$. Right away, notice that $|\mathcal{O}_{D_i}| < |\mathcal{O}_D|$ since otherwise D would also be active by the second part of the definition. But then

$$\mathsf{drad}(D) \leq x \leq \mathsf{drad}(D_i) + 2 \leq 4|\mathcal{O}_{D_i}| + 3 < 4(|\mathcal{O}_{D_i}| + 1) \leq 4|\mathcal{O}_D|$$

contradicts the assumption of D being inactive. As such, we know that all D_i ($i = 1, \ldots, \ell$) must be inactive or have radius less than $x-2$. In other words, each inactive cluster in $\mathcal{C}_x$ descends from the set

$$\{D \in \mathcal{C}_{x-1} \mid D \text{ is inactive}\} \cup \{D \in \mathcal{C}_{x-1} \mid \mathsf{drad}(D) < x - 2\}.$$

The cardinality of the set on the left is m_{x-1} and, by Lemma 3.2.5, the cardinality of the set on the right is at most k. This proves the claim. □

Corollary 3.2.11. *$\mathcal{C}_{4k+2}$ consists of at most $2^k + 4k^2 + k$ clusters.*

Notice that Theorem 3.2.4 is an immediate consequence of Corollary 3.2.11 and Lemma 3.2.7.

An Upper Bound for Diameter-Based Cost

The main challenge in proving an upper bound on the approximation guarantee of complete linkage when replacing the drad objective by the diam objective is to deal with the possibly large increase of cost after a merge step. When we perform complete linkage for the drad objective, complete linkage roughly halves the number of clusters while the radius increases by a constant amount and this is repeated as long as the number of clusters is larger than k (see Corollary 3.2.6). This is an easy conclusion from the fact that whenever the centers of two clusters are contained in the same optimal cluster merging the two clusters increases the radius only by twice the radius of the optimal clustering. If we try to apply this insight to analyze complete linkage for the diam objective we now consider the merge of two clusters which intersect the same optimal cluster. As we see in Figure 3.10 merging these two clusters can double the diameter in the worst case, therefore we are not able to prove a similar statement to Corollary 3.2.6 for diam. In conclusion when dealing with the diameter we ignore Part 1 of the analysis for the radius and instead follow some ideas of Part 2 where we divide clusters constructed by complete linkage in active and inactive clusters. Even though we substantially change the definition of inactive and active clusters the main idea stays the same: the diameter

of active clusters can be upper bounded nicely while we guarantee that there are not too many inactive clusters. A main difference to Part 2 of the analysis for the radius is now that the total number of active and inactive clusters must be upper bounded by k instead of $O(2^k)$, which yields the increase in the approximation factor for diameter.

We now give a brief overview over the ideas used to upper bound the approximation factor of complete linkage for the diameter. For some arbitrary but fixed k let $\mathcal{O}$ denote an optimal diam solution of size k and assume that $\mathsf{diam}(\mathcal{O}) = 1$ from now on. Consider the clustering $\mathcal{C}_1$ computed by complete linkage at time $t_{\leq 1}$. Observe that every optimal cluster can fully contain at most one cluster from $\mathcal{C}_1$, as the union of such clusters would cost at most 1. Now, consider the graph $G = (V, E)$ with $V = \mathcal{O}$ and edges $\{A, B\} \subset V$ for every cluster $C \in \mathcal{C}_1$ intersecting A and B. If there is such an edge $\{A, B\}$, then the cost of merging A and B is upper bounded by 3. We can go even further and consider the merge of all optimal clusters in a connected component of G. Suppose the size of the connected component is m, then the resulting cluster costs at most $2m - 1$. There are two extreme cases in which we could end up: if $E = \emptyset$, then $\mathcal{C}_1 = \mathcal{O}$ and complete linkage has successfully recovered the optimal solution. On the other hand, if G is connected, then merging all points costs at most $2k - 1$ and we get an $O(k)$-approximative solution. The remaining cases are more difficult to handle. We proceed by successively adding edges between optimal clusters, while maintaining the property that for a connected component Z in G merging $\cup_{A \in V(Z)} A$ costs at most $|V(Z)|^{\ln(3)/\ln(2)}$. This leads to an upper bound of $\lceil k^{\ln(3)/\ln(2)} \rceil$ for all clusters C constructed by complete linkage with $C \subset \cup_{A \in V(Z)} A$. We call such clusters *active clusters*. All clusters which do not admit this property are called *inactive clusters*. We show that the number of inactive clusters is sufficiently small, such that in the end, we are able to prove that $\mathcal{C}_{\lceil k^{\ln(3)/\ln(2)} \rceil}$ consists of at most k clusters. This immediately leads the following theorem.

Theorem 3.2.12. *Let $\mathscr{H} = (\mathcal{H}_n, \ldots, \mathcal{H}_1)$ be the hierarchical clustering computed by complete linkage on $(\mathcal{P}, d)$ with respect to* diam*. For all $1 \leq k \leq n$ the diameter $\mathsf{diam}(\mathcal{H}_k)$ is upper bounded by $\lceil k^{\ln(3)/\ln(2)} \rceil \mathsf{diam}(\mathcal{O}_k)$, where $\mathcal{O}_k$ is an optimal k-clustering with respect to* diam*.*

Let $\alpha = \ln(3)/\ln(2)$ from now on. Essential for this section is a sequence of *cluster graphs* $G_t = (V_t, E_t)$ for $t = 1, \ldots, \lceil k^\alpha \rceil$ constructed directly on the set $V_t = \mathcal{O}$ of optimal k-clusters. We start with the cluster graph G_1 that contains edges $\{A, B\}$ for every two vertices $A, B \in V_1 = \mathcal{O}$ that are hit by a common cluster from $\mathcal{C}_1$. To this we successively add edges based on a vertex labeling in order to create the remaining cluster graphs $G_2, \ldots, G_{\lceil k^{\ln(3)/\ln(2)} \rceil}$. The labeling distinguishes vertices as being either *active* or *inactive*. We denote the set of active vertices in V_t by V_t^a and the set of inactive ones by V_t^i. In the beginning ($t = 1$) the inactive vertices are set to precisely those that are isolated: $V_1^i = \{O \in V_1 \mid \delta_{G_1}(O) = \emptyset\}$. For $t \geq 2$, the labeling is outlined in Definition 3.2.13. Over the course of time, active vertices may become inactive, but inactive vertices never become active again.

Given a labeling for V_{t+1}, we construct G_{t+1} from G_t by adding additional edges: If there are two active vertices $A, B \in V_{t+1}^a$ that are both hit by a common cluster from $\mathcal{C}_{t+1}$, we add an edge $\{A, B\}$ to E_{t+1}.

Definition 3.2.13. *Let $t \geq 1$ and $A \in V_{t+1}$ be an arbitrary optimal cluster and Z_A the connected component in G_t that contains A. We call A* inactive *(i.e., $A \in V_{t+1}^i$) if $\lceil \mathsf{diam}(Z_A) \rceil \leq t$, and* active *otherwise. Here, and in the following $\mathsf{diam}(Z_A) = \mathsf{diam}(\bigcup_{B \in V(Z_A)} B)$ denotes the cost of merging all optimal clusters contained in $V(Z_A)$.*

Thus if a connected component in G_t has small diameter, then all vertices in this component become inactive in G_{t+1} by definition. We state the following useful properties of inactive vertices in $(G_t)_{t=1}^{\lceil k^\alpha \rceil}$.

Lemma 3.2.14. *If Z is a connected component in G_{t+1} with $V(Z) \cap V_{t+1}^i \neq \emptyset$, then*

1. *Z is also a connected component in G_t and $\lceil \mathsf{diam}(Z) \rceil \leq t$,*
2. *we have $V(Z) \subseteq V_{t+1}^i$, i.e., all vertices in Z become inactive at the same time.*

Moreover we have $V_t^i \subseteq V_{t+1}^i$, so once vertices become inactive, they stay inactive. Equivalently, $V_{t+1}^a \subseteq V_t^a$.

Proof. Take any inactive vertex $A \in V_{t+1}^i \cap V(Z)$ and consider the connected component Z_A in G_t containing A. By Definition 3.2.13, we have that $\lceil \mathsf{diam}(Z_A) \rceil \leq t$ and so all other vertices in Z_A have to be in V_{t+1}^i as well. We observe that $E_{t+1} \setminus E_t$ only contains edges between vertices from V_{t+1}^a by construction. This shows $Z = Z_A$.

It is left to show that inactive vertices stay inactive. For $t = 1$ the inactive vertices V_1^i are already connected components with diameter at most 1. As such, they remain inactive at step $t = 2$. For $t \geq 2$, consider an inactive vertex $A \in V_t^i$ and the connected component $Z \subseteq G_t$ containing it. We showed previously that $V(Z) \subset V_t^i$ and so Z is also a connected component in G_{t+1} with $\lceil \mathsf{diam}(Z) \rceil \leq t - 1 < t$ and thus $A \in V(Z) \subset V_{t+1}^i$. □

Definition 3.2.15. *Let $C \in \mathcal{C}_t$ for some fixed $t \in \mathbb{N}$. We define $\mathscr{I}_t = \{C \in \mathcal{C}_t \mid \mathcal{O}_C \cap V_t^i \neq \emptyset\}$ as the set of all clusters in $\mathcal{C}_t$ which hit at least one inactive vertex of G_t. We call these clusters inactive and all clusters from $\mathcal{C}_t \setminus \mathscr{I}_t$ active.*

We prove the following easy property about active clusters.

Lemma 3.2.16. *If $C \in \mathcal{C}_t \setminus \mathscr{I}_t$, then $G_t[\mathcal{O}_C]$ forms a clique. In particular there exists a connected component in G_t that fully contains $\mathcal{O}_C$.*

Proof. By definition of $\mathscr{I}_t$, $\mathcal{O}_C$ must consist exclusively of active vertices. Since all of them are hit by $C \in \mathcal{C}_t$ there exists an edge $\{A, B\} \in E_t$ for every pair $A, B \in \mathcal{O}_C$. In other words, $G_t[\mathcal{O}_C]$ forms a clique and the claim follows. □

This does not necessarily hold for an inactive cluster $C \in \mathscr{I}_t$. As C contains at least one inactive vertex, the connected component Z which contains this vertex does not grow. If later on complete linkage merges C with another cluster the result is an inactive cluster which may hit vertices outside of Z. So $G_{t'}$ does not reflect the progression of C for $t' \geq t$. However, the number of such clusters cannot exceed $|V_t^i|$.

Lemma 3.2.17. *The number of inactive clusters in $\mathcal{C}_t$ is at most the number of inactive vertices at time t. That is, $|\mathscr{I}_t| \leq |V_t^i|$ holds for all $t \in \mathbb{N}$.*

Proof. We prove the claim by showing that the following inductive construction defines a family of injective mappings $\phi_t : \mathscr{I}_t \to V_t^i$:

- Let $C \in \mathscr{I}_1$ be an inactive cluster. By definition C thus has to intersect an inactive optimal cluster $A \in V_1^i$. Actually, there can only be one such cluster, as any other optimal cluster that is hit would induce an edge incident to A in G_1, making it active. Set $\phi_1(C) = A$, so that $\mathcal{O}_C = \{\phi_1(C)\}$.

- For $t > 1$ and $C \in \mathscr{I}_t$ we distinguish two cases: If there is no cluster in $\mathscr{I}_{t-1}$ that is a subset of C, we pick an arbitrary but fixed $A \in \mathcal{O}_C \cap V_t^i$ and set $\phi_t(C) = A$. Otherwise, we know that C must descend from some cluster $D \in \mathscr{I}_{t-1}$ and we can set $\phi_t(C) = \phi_{t-1}(D)$. Since $\phi_{t-1}(D) \in V_{t-1}^i \subset V_t^i$ by Lemma 3.2.14, this shows that ϕ_t really maps into V_t^i.

Suppose that there exist two inactive clusters $C, D \in \mathscr{I}_1$ that are mapped to the same inactive vertex $A \in V_1^i$. Then, by the construction of ϕ_1, $\mathcal{O}_C = \{A\} = \mathcal{O}_D$ shows that C and D are actually fully contained in the same optimal cluster. The optimal cluster has diameter at most 1 and so C and D would have already been merged in $\mathcal{C}_1$. As this is not possible, ϕ_1 has to be injective.

Now, let $t \geq 2$ be arbitrary and assume ϕ_{t-1} to be injective. We show that in that case ϕ_t also has to be injective. Suppose on the contrary that there exist two different clusters $C, D \in \mathscr{I}_t$ with $\phi_t(C) = \phi_t(D)$. We distinguish three cases.

Case 1: Both C and D descend from (i.e., contain) clusters $C', D' \in \mathscr{I}_{t-1}$ with $\phi_t(C) = \phi_{t-1}(C')$ and $\phi_t(D) = \phi_{t-1}(D')$, respectively. Then $\phi_{t-1}(C') = \phi_t(C) = \phi_t(D) = \phi_{t-1}(D')$ entails that $C' = D'$, since ϕ_{t-1} is assumed to be injective. Clearly, $C' = D'$ cannot end up being a subset of two different clusters in $\mathscr{I}_t$ and so we end up in a contradiction.

Case 2: Neither C nor D descend from a cluster in $\mathscr{I}_{t-1}$. In other words, C and D fully descend from clusters in $\mathcal{C}_{t-1} \setminus \mathscr{I}_{t-1}$ and so there exist clusters $C', D' \in \mathcal{C}_{t-1} \setminus \mathscr{I}_{t-1}$ contained in C and D, respectively, such that $A = \phi_t(C) = \phi_t(D) \in \mathcal{O}_{C'} \cap \mathcal{O}_{D'}$. Applying Lemma 3.2.16 yields the existence of a connected component Z in G_{t-1} with $V(Z) \supset \mathcal{O}_{C'} \cup \mathcal{O}_{D'}$. We show that this connected component has diameter at most $t-1$. In that case, C' and D' should have already been merged in $\mathcal{C}_{t-1}$; a contradiction. To show that $\mathsf{diam}(Z) \leq t-1$, consider the connected component Z' in G_t containing $A = \phi_t(C) = \phi_t(D) \in \mathcal{O}_C \cap \mathcal{O}_D \cap V_t^i$. Since A was chosen from a subset of V_t^i, we know from Lemma 3.2.14 that Z' is also a connected component in G_{t-1} with $\mathsf{diam}(Z') \leq t-1$. Now, $A \in V(Z) \cap V(Z')$ shows that $Z = Z'$ and so we are done.

Case 3: D contains a cluster $D' \in \mathscr{I}_{t-1}$, so that $\phi_t(D) = \phi_{t-1}(D') \in V_{t-1}^i$, whereas C does not. (The symmetric case with the roles of C and D swapped is left out.) Since C fully descends from $\mathcal{C}_{t-1} \setminus \mathscr{I}_{t-1}$, we know that $\mathcal{O}_C \subseteq V_{t-1}^a$. But this already yields a contradiction: $V_{t-1}^a \ni \phi_t(C) = \phi_t(D) = \phi_{t-1}(D') \in V_{t-1}^i$.

This covers all possible cases, with each one ending in a contradiction. Hence ϕ_t has to be injective and by induction this holds for all $t \in \mathbb{N}$. □

Active clusters from $\mathcal{C}_t$ are nicely represented by the graph G_t as shown in Lemma 3.2.16. We can indirectly bound the diameter of active clusters by bounding the diameter of the connected components they are contained in.

Lemma 3.2.18. *Let Z be a connected component in G_t. If $V(Z) \subset V_t^a$, we have $\lceil \mathsf{diam}(Z) \rceil \leq |V(Z)|^\alpha$.*

Proof. Again, we prove this via an induction over t. For $t = 1$ and $A, B \in V(Z)$ we want to upper bound the distance between $p \in A$ and $q \in B$. Let $A = Q_1, \ldots, Q_s = B$ be a simple path connecting A and B in Z. We know by definition of G_1 that for

$j = 1, \dots, s-1$ there is a pair of points $p_j \in Q_j$ and $q_j \in Q_{j+1}$ with $d(p_j, q_j) \leq 1$. Using the triangle inequality we obtain

$$\begin{aligned} d(p,q) &\leq d(p,p_1) + \sum_{j=1}^{s-2} \Big(d(p_j,q_j) + d(q_j,p_{j+1}) \Big) \\ &\quad + d(p_{s-1},q_{s-1}) + d(q_{s-1},q) \\ &\leq 2s-1. \end{aligned}$$

Here we use that q_j and p_{j+1} are in the same optimal cluster, thus the distance between those points is at most one.

Since $V(Z)$ contains only active vertices we have $|V(Z)| \geq 2$. Using the above upper bound on the distance between two points in $\bigcup_{A \in V(Z)} A$ we obtain

$$\lceil \mathsf{diam}(Z) \rceil \leq 2|V(Z)| - 1 \leq |V(Z)|^\alpha,$$

where the last inequality follows from the fact that the function $h(x) = x^\alpha - 2x + 1$ is convex and $h(1) = h(2) = 0$. Thus $h(x) \leq 0$ for $x \in (1,2)$ and $h(x) \geq 0$ for $x \in \mathbb{R}_{\geq 0} \backslash (1,2)$.

For $t > 1$ let $Z_1, \dots, Z_u$ denote the connected components in G_{t-1} with $V(Z) = \bigcup_{j=1}^u V(Z_j)$. Let $j \in \{1, \dots, u\}$. We observe that $V(Z_j) \subset V(Z) \subset V_t^a \subset V_{t-1}^a$. Thus we obtain by induction that

$$\lceil \mathsf{diam}(Z_j) \rceil \leq |V(Z_j)|^\alpha. \tag{3.2}$$

Suppose that $\lceil \mathsf{diam}(Z_j) \rceil \leq t-1$. Then $V(Z_j) \subset V_t^i$ by definition, which is a contradiction to $V(Z) \cap V_t^i = \emptyset$. So we must have

$$t \leq \lceil \mathsf{diam}(Z_j) \rceil. \tag{3.3}$$

Combining (3.2) and (3.3) we obtain

$$t \leq |V(Z_j)|^\alpha. \tag{3.4}$$

For $A, B \in V(Z)$ we want to upper bound the distance between $p \in A$ and $q \in B$. Let $A = Q_1, \dots, Q_s = B$ be a simple path connecting A and B in Z which enters and leaves every connected component Z_j for $j \in \{1, \dots, u\}$ at most once. We divide the path into several parts such that every part lies in one connected component from $\{Z_1, \dots, Z_u\}$. Let $1 = m_1 < m_2 < \ldots < m_\ell = s$ such that $Q_{m_j} \dots, Q_{m_{j+1}-1}$ lie in one connected component $Z^{(j)} \in \{Z_1, \dots, Z_u\}$ and $Z^{(j)} \neq Z^{(j+1)}$ for all $j \in \{1, \dots, \ell\}$. Since $(Q_{m_j - 1}, Q_{m_j}) \in E_t$ we know that there exists a cluster in $\mathcal{C}_t$ that intersects $Q_{m_j - 1}$ and Q_{m_j}, thus there is a pair of points $p_j \in Q_{m_j - 1}$ and $q_j \in Q_{m_j}$ such that $d(p_j, q_j) \leq t$. We obtain

$$\begin{aligned} \lceil d(p,q) \rceil &\leq \sum_{j=1}^{\ell-1} \Big(\lceil \mathsf{diam}(Z^{(j)}) \rceil + \lceil d(p_j,q_j) \rceil \Big) + \lceil \mathsf{diam}(Z^{(\ell)}) \rceil \\ &\leq (\ell-1)t + \sum_{j=1}^{\ell} |V(Z^{(j)})|^\alpha \\ &\leq (\ell-1) \min_{1 \leq j \leq \ell} |V(Z^{(j)})|^\alpha + \sum_{j=1}^{\ell} |V(Z^{(j)})|^\alpha. \end{aligned}$$

For the second inequality we use (3.2) and $d(p_j, q_j) \leq t$. For the third inequality we use (3.4). Now it remains to show that

$$(\ell - 1) \min_{1 \leq j \leq \ell} |V(Z^{(j)})|^\alpha + \sum_{j=1}^{\ell} |V(Z^{(j)})|^\alpha \leq \Big(\sum_{j=1}^{\ell} |V(Z^{(j)})| \Big)^\alpha.$$

For this purpose we assume without loss of generality that $|V(Z^{(1)})| \geq |V(Z^{(2)})| \geq \ldots \geq |V(Z^{(\ell)})|$ and define $x_j = \frac{|V(Z^{(j)})|}{|V(Z^{(1)})|}$ for $j = 1, \ldots, \ell$. We obtain the following equivalent inequality:

$$|V(Z^{(1)})|^\alpha \Big((\ell - 1) x_\ell^\alpha + \sum_{j=1}^{\ell} x_j^\alpha \Big) \leq |V(Z^{(1)})|^\alpha \Big(\sum_{j=1}^{\ell} x_j \Big)^\alpha.$$

Since $1 = x_1 \geq \ldots \geq x_\ell \geq 0$ this follows directly from Lemma 3.2.19 and thus

$$\lceil d(p, q) \rceil \leq \Big(\sum_{j=1}^{\ell} |V(Z^{(j)})| \Big)^\alpha \leq |V(Z)|^\alpha.$$

We obtain $\lceil \mathsf{diam}(Z) \rceil \leq |V(Z)|^\alpha$ which proves the lemma. □

Lemma 3.2.19. *For $\ell \geq 1$ let*

$$f(y_1, \ldots, y_\ell) = \Big((\ell - 1) y_\ell^\alpha + \sum_{j=1}^{\ell} y_j^\alpha \Big) - \Big(\sum_{j=1}^{\ell} y_j \Big)^\alpha.$$

We have $f(y_1, \ldots, y_\ell) \leq 0$ for all $1 = y_1 \geq y_2 \geq \ldots \geq y_\ell \geq 0$.

Proof. Let $M = \{(a_1, \ldots, a_\ell) \mid 1 = a_1 \geq a_2 \geq \ldots \geq a_\ell \geq 0\}$. We have to prove that $f(y_1, \ldots, y_\ell) \leq 0$ for all $(y_1, \ldots, y_\ell) \in M$.

Let $s \geq 1$ and consider all points in M whose sum of coordinates is exactly s, i.e., $M_s = \{(a_1, \ldots, a_\ell) \in M \mid a_1 + \ldots + a_\ell = s\}$. Notice that M_s is a convex polytope and f is a convex function on M_s and thus the maximum of f in M_s is attained at one of the vertices of M_s. These vertices have the form $(y_1, \ldots, y_\ell)$ of f in M_s must be of the form $y_1 = \ldots = y_{\ell_1} = 1$ and $y_{\ell_1 + 1} = \ldots = y_\ell = b$. We obtain for $k = \ell - \ell_1$

$$\begin{aligned} f(y_1, \ldots, y_\ell) &= (\ell - 1) b^\alpha + \ell_1 + (\ell - \ell_1) b^\alpha - (\ell_1 + (\ell - \ell_1) b)^\alpha \\ &= \ell_1 + (2\ell - \ell_1 - 1) b^\alpha - (\ell_1 + (\ell - \ell_1) b)^\alpha \\ &= \ell_1 + (2k + \ell_1 - 1) b^\alpha - (\ell_1 + kb)^\alpha. \end{aligned}$$

It is left to show that for all $b \in [0, 1]$ and natural numbers $\ell_1 \geq 1$, $k \geq 0$ this term is at most 0. Thus we define for $\ell_1, k, b \in \mathbb{R}_{\geq 0}$ the function

$$g(\ell_1, k, b) = \ell_1 + (2k + \ell_1 - 1) b^\alpha - (\ell_1 + kb)^\alpha.$$

The partial derivative of g with respect to k is given by

$$\frac{\partial}{\partial k} g(\ell_1, k, b) = 2b^\alpha - \alpha b (\ell_1 + bk)^{\alpha - 1}$$

Now suppose that either $\ell_1 \geq 2$ and $b \in [0,1]$ or $\ell_1 = 1, k \geq 1$ and $b \in [0,1]$. In both cases we obtain

$$b < 0.67(\ell_1 + bk) \leq \left(\frac{\alpha}{2}\right)^{\frac{1}{\alpha-1}} (\ell_1 + bk)$$

and thus

$$\frac{\partial}{\partial k} g(\ell_1, k, b) < 2b\left(\frac{\alpha}{2}\right)^{\frac{\alpha-1}{\alpha-1}} (\ell_1 + bk)^{\alpha-1} - \alpha b(\ell_1 + bk)^{\alpha-1} = 0$$

Therefore g is monotonically decreasing for these values and we conclude that the maximum of g for $b \in [0,1]$ and natural numbers $\ell_1 \geq 1, k \geq 0$ must be attained at one of the points $(\ell_1, 0, b), (1, 1, b)$ for $\ell_1 \in \mathbb{N}_{\geq 1}, b \in [0,1]$. Now

$$g(\ell_1, 0, b) = \ell_1 + (\ell_1 - 1)b^\alpha - \ell_1^\alpha$$

is monotonically increasing in b, so the maximum is attained for $b = 1$. Observe that $h(\ell_1) = g(\ell_1, 0, 1) = 2\ell_1 - 1 - \ell_1^\alpha$ in concave and we have $h(1) = h(2) = 0$. Thus $h(\ell_1) \geq 0$ for $\ell_1 \in (1,2)$ and $h(\ell_1) \leq 0$ for $\ell_1 \in \mathbb{R}_{\geq 0} \backslash (1,2)$.

It is left to check the value of g at $(1,1,b)$ for $b \in [0,1]$. Let $\phi(b) = g(1,1,b) = 1 + 2b^\alpha - (1+b)^\alpha$. We consider the second derivative of ϕ which is given by

$$\frac{\mathrm{d}^2}{\mathrm{d}b^2}\phi(b) = \alpha(\alpha-1)(2b^{\alpha-2} - (1+b)^{\alpha-2}).$$

Since $b^{\alpha-2} \geq (1+b)^{\alpha-2}$ we obtain that $\frac{\mathrm{d}^2}{\mathrm{d}b^2}\phi(b) \geq 0$ and therefore $\frac{\mathrm{d}}{\mathrm{d}b}\phi(b)$ is monotonically increasing. This implies that ϕ is convex on $[0,1]$. Since $\phi(0) = \phi(1) = 0$ and ϕ is convex we know that $\phi(b) \leq 0$ for all $b \in [0,1]$. This proves the lemma. □

We see that a connected component in $G_{\lceil k^\alpha \rceil}$ cannot contain two active clusters, yielding the following upper bound.

Lemma 3.2.20. *At time $t_{\leq \lceil k^\alpha \rceil}$ the number of active clusters is less than or equal to the number of active vertices. In other words, $|\mathcal{C}_{\lceil k^\alpha \rceil} \setminus \mathscr{I}_{\lceil k^\alpha \rceil}| \leq |V^a_{\lceil k^\alpha \rceil}|$.*

Proof. By Lemma 3.2.16 we know that for every cluster $C \in \mathcal{C}_{\lceil k^\alpha \rceil} \setminus \mathscr{I}_{\lceil k^\alpha \rceil}$ the set $\mathcal{O}_C$ is fully contained in a connected component Z_C from $G_{\lceil k^\alpha \rceil}$. We show that mapping any such C to an arbitrary vertex in Z_C yields an injective map $\varphi : \mathcal{C}_{\lceil k^\alpha \rceil} \setminus \mathscr{I}_{\lceil k^\alpha \rceil} \hookrightarrow V^a_{\lceil k^\alpha \rceil}$. First, notice that φ is well-defined: If Z_C contains an inactive vertex, then all its vertices are inactive (Lemma 3.2.14), contradicting the choice of C as active.

Suppose now that there are two different clusters $C, C' \in \mathcal{C}_{\lceil k^\alpha \rceil} \setminus \mathscr{I}_{\lceil k^\alpha \rceil}$ that are mapped to the same vertex $\varphi(C) = \varphi(C')$. Then the connected components Z_C and $Z_{C'}$, in which they are embedded, already have to coincide ($Z_C = Z_{C'}$). But we have just shown (Lemma 3.2.18), that $\mathsf{diam}(Z_C) \leq |V(Z_C)|^{\ln(3)/\ln(2)} \leq \lceil k^\alpha \rceil$ and so C and C' would have already been merged in $\mathcal{C}_{\lceil k^\alpha \rceil}$. As such the images of both cannot coincide and the map is injective. □

Together with the bound for the number of inactive clusters we are now able to prove the theorem.

Prroof of Theorem 3.2.12. Using Lemma 3.2.17 and 3.2.20 we obtain $|\mathcal{C}_{\lceil k^\alpha \rceil}| = |\mathcal{C}_{\lceil k^\alpha \rceil} \setminus \mathscr{I}_{\lceil k^\alpha \rceil}| + |\mathscr{I}_{\lceil k^\alpha \rceil}| \leq |V^a_{\lceil k^\alpha \rceil}| + |V^i_{\lceil k^\alpha \rceil}| = k$, yielding $\mathsf{diam}(\mathcal{H}_k) \leq \mathsf{diam}(\mathcal{C}_{\lceil k^\alpha \rceil}) \leq \lceil k^\alpha \rceil \mathsf{diam}(\mathcal{O}_k)$. □

3.2.3 A Lower Bound for Complete Linkage

In the following we show that complete linkage performs asymptotically bad. That is, for every $k \in \mathbb{N}$ we provide an instance $(V(P_k), d)$ on which the diameter and radius of a k-clustering computed by complete linkage is off by a factor of $\Omega(k)$ from the cost of an optimal solution. This improves upon the previously known lower bound of $\Omega(\log_2(k))$ established by Dasgupta and Long [39]. Recall that one of the big problems preventing an improved lower bound was that any horizontal merge already paid for all the involved rows. As such, for the worst case one was only allowed to merge vertically, but this can be done at most $\log_2(k)$ times. We improve upon this by inductively constructing an instance from smaller components that are diagonally shifted to produce bigger ones. Merging two such diagonally shifted components incurs an additional cost of 1, while ensuring at the same time that this does not pay for any future merges of parallel components.

A k-component $K_k = (G_k, \phi_k)$ is a combination of a graph $G_k = (V_k, E_k)$ and a mapping $\phi_k : V_k \to \{1, \dots, k\}$. The mapping is necessary for the construction of the component and later on determines an optimal k-clustering on P_k. We refer to $\phi_k(x)$ as the *level* of x. The other part of the component is an undirected graph G_k, referred to as a *k-graph*, on 2^{k-1} points with edge weights in $\mathbb{N}$ that describe the distances between the levels.

The 1-component K_1 consists of a single point x with $\phi_1(x) = 1$. All higher components are constructed inductively from this 1-component. Given the $(k-1)$-component K_{k-1} we construct K_k as follows: Let $K_{k-1}^{(0)}$ and $K_{k-1}^{(1)}$ be two copies of the $(k-1)$-component K_{k-1}. For the k-graph G_k we first take the disjoint union of the graphs $G_{k-1}^{(0)}$ and $G_{k-1}^{(1)}$. This already yields all the points of G_k. For the k-mapping ϕ_k we set $\phi_k(x) = \phi_{k-1}^{(i)}(x) + i$ for $x \in V(G_{k-1}^{(i)}) \subset V(G_k)$. That is, in the first copy the levels stay the same, whereas in the second all levels are shifted by 1. Finally, to complete G_k, we add one edge of weight $k-1$ from the unique point $s \in V(G_k)$ with $\phi_k(s) = 1$ to the unique point $t \in V(G_k)$ with $\phi_k(t) = k$. The progression of the first five components is given in Figure 3.12.

The instance $(V(P_k), d)$ is now constructed from the k-component as follows: Let $K_k^{(1)}, \dots, K_k^{(k+1)}$ be $k+1$ copies of K_k. Take the disjoint union of the corresponding k-graphs $G_k^{(1)}, \dots, G_k^{(k+1)}$ and connect them by adding edges $\{x, y\}$ of weight 1 for every two points $x \in V(G_k^{(i)})$ and $y \in V(G_k^{(j)})$ with $\phi_k^{(i)}(x) = \phi_k^{(j)}(y)$. Note that the sets of points from the same level constitute cliques of diameter and radius 1 and form an optimal solution of cost 1. To simplify notation we omit the indices and write $\phi_k(x)$ to denote the level of a point $x \in V(G_k^{(j)}) \subset V(P_k)$. The distance between two points in $V(P_k)$ is given by the length of a shortest path.

Let $(\mathcal{H}_{k'})_{k'=1}^{n}$ be the clustering produced by complete linkage on $(V(P_k), d)$ minimizing the radius or diameter. Recall that $\mathcal{H}_{k'-1}$ arises from $\mathcal{H}_{k'}$ by merging two clusters $A, B \in \mathcal{H}_{k'}$ that minimize the radius or diameter of $A \cup B$. Remember that for $\mathsf{cost} \in \{\mathsf{diam}, \mathsf{drad}\}$ we define $t_{\leq x} = \min\{k' \mid \mathsf{cost}(\mathcal{H}_{k'}) \leq x\}$ and that $\mathcal{C}_x = \mathcal{H}_{t_{\leq x}}$ denotes the smallest clustering with cost smaller or equal to x. We show in the following two subsections that $\mathcal{C}_{k-1}$ consists exactly of the $k+1$ different k-graphs that make up the instance resulting in the following theorem.

Theorem 3.2.21. *For every $k \in \mathbb{N}$ there exists an instance $(V(P_k), d)$ on which com-*

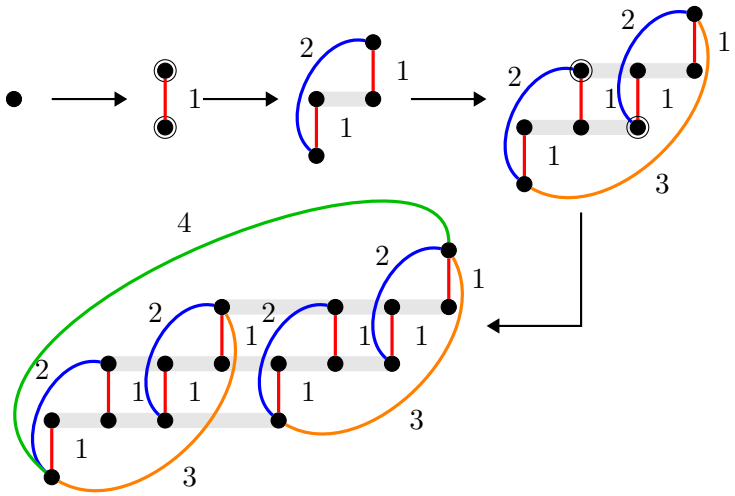

Figure 3.12: The progression of the first 5 components $K_1, \ldots, K_5$. The gray sets indicate points on the same level and form the optimal clusters. When analyzing the instance for the radius, the encircled points in the K_2 and K_4 component indicate their optimal centers.

plete linkage, minimizing either diam *or* drad*, computes a solution of diameter k or radius $\frac{k}{2}$, respectively, whereas the cost of an optimal solution is* 1*.*

A Lower Bound for Diameter-Based Cost

We start with the analysis for diameter-based costs and after that move on to radius-based costs.

Lemma 3.2.22. *The distance between two points $x, y \in V(P_k)$ is at least as big as the difference in levels $|\phi_k(x) - \phi_k(y)|$.*

Proof. By the inductive construction of the components, an edge of weight w can cross at most w levels. Hence the distance between x and y is at least $|\phi_k(x) - \phi_k(y)|$. □

Consider an ℓ-graph G_ℓ. Instead of talking about the cluster $V(G_\ell)$ in $(V(P_k), d)$ we slightly abuse our notation and see G_ℓ as a cluster with $\mathsf{diam}(G_\ell) = \max_{x,y \in V(G_l)} d(x, y)$, i.e., the diameter of $V(G_\ell)$. Using the previous lemma we can show inductively that the diameter of any ℓ-graph in P_k is $\ell - 1$.

Lemma 3.2.23. *Let G_ℓ be an ℓ-graph contained in P_k. We have $\mathsf{diam}(G_\ell) = \ell - 1$.*

Proof. We prove the upper bound $\mathsf{diam}(G_\ell) \leq \ell - 1$ by induction. The 1-graphs are points and so the claim follows trivially for $\ell = 1$. Assume now that we have shown the claim for $\ell - 1$. Let $s, t \in V(G_\ell)$ be points such that $d(s, t) = \mathsf{diam}(G_\ell)$. If these points lie in the same graph, say $G^{(0)}_{\ell-1}$, of the two $(\ell-1)$-graphs $G^{(0)}_{\ell-1}$ and $G^{(1)}_{\ell-1}$ that make up G_ℓ, then

$$\mathsf{diam}(G_\ell) = d(s, t) \leq \mathsf{diam}(G^{(0)}_{\ell-1}) \leq \ell - 2 < \ell - 1$$

by induction and we are done. Otherwise we may assume that $s \in V(G^{(0)}_{\ell-1})$ and $t \in V(G^{(1)}_{\ell-1})$. This leaves us with another case analysis. If s is the unique point with level 1

and t is the unique point in level ℓ in G_ℓ then we are again done, since by construction there exists an edge between s and t of weight $\ell - 1$. Otherwise one of s or t must share a level with a point not in the same $(\ell-1)$-graph as themselves. Without loss of generality we may assume that s lies in the same level as some $u \in V(G^{(1)}_{\ell-1})$. By induction $d(u,t) \leq \ell - 2$ and so

$$\mathsf{diam}(G_\ell) = d(s,t) \leq d(s,u) + d(u,t) \leq 1 + \ell - 2 = \ell - 1.$$

This concludes the proof of the upper bound $\mathsf{diam}(G_\ell) \leq \ell - 1$.

To see the lower bound $\mathsf{diam}(G_\ell) \geq \ell - 1$, we apply Lemma 3.2.22 to the unique point s with level 1 and the unique point t with level ℓ in G_ℓ. This shows that $\mathsf{diam}(G_\ell) \geq d(s,t) \geq \ell - 1$. □

The goal now is to show that complete linkage actually reconstructs these graphs as clusters. We already computed the diameter of an ℓ-graph and now it is left to observe that merging two ℓ-graphs costs at least ℓ.

Lemma 3.2.24. *Complete linkage might merge clusters on $(V(P_k), d)$ in such a way that for all $\ell \leq k$, the clustering $\mathcal{C}_{\ell-1}$ consists exactly of the ℓ-graphs that make up P_k.*

Proof. We again prove the claim by induction. Complete linkage always starts with every point in a separate cluster. Since those are exactly the 1-graphs and any merge costs at least 1, the claim follows for $\ell = 1$. Suppose now that $\mathcal{C}_{\ell-1}$ consists exactly of the ℓ-graphs of the instance. Since we are dealing with integer weights, any new merge increases the diameter by at least 1 and so we may merge all pairs of ℓ-graphs that form the $(\ell+1)$-graphs. These are cheapest merges as they altogether increase the diameter from $\ell - 1$ to ℓ (see Lemma 3.2.23). To finish the proof we are left to show that at this point there are no more free merges left. Take any two $(\ell+1)$-graphs $G_{\ell+1} \neq G'_{\ell+1}$ contained in the current clustering. If they do not exactly cover the same levels, then the distance between the point in the lowest level to the point in the highest level is strictly more than ℓ by Lemma 3.2.22. Hence, we can assume that they share the same levels, say level λ up to level $\ell + \lambda$. Denote by s the unique point in $V(G_{\ell+1})$ with $\phi_k(s) = \lambda$ and by t the unique point in $V(G'_{\ell+1})$ with $\phi_k(t) = \ell + \lambda$. A shortest path connecting s and t must contain an edge $\{u, w\}$ with $u \in V(G_{\ell+1})$ and $w \in V(P_k) \backslash V(G_{\ell+1})$. Such an edge either weights at least $\ell + 1$ or weights 1 and connects points in the same level, i.e., $\phi_k(u) = \phi_k(w)$. In the first case we directly obtain $d(s,t) \geq \ell + 1$. In the second case we use Lemma 3.2.22 and obtain

$$\begin{aligned} d(s,t) &= d(s,u) + d(u,w) + d(w,t) \\ &\geq |\phi_k(s) - \phi_k(u)| + 1 + |\phi_k(w) - \phi_k(t)| \\ &= |\phi_k(s) - \phi_k(t)| + 1 \\ &= \ell + 1. \end{aligned}$$

It follows that $\mathcal{C}_\ell$ consists exactly of the $(\ell+1)$-graphs that make up P_k. □

Proof of Theorem 3.2.21 (diameter). Lemma 3.2.24 shows that $\mathcal{C}_{k-1}$ can consist of all the k-graphs that make up P_k. There are exactly $k+1$ of them and so there is one merge remaining to get a k-clustering. By definition of $\mathcal{C}_{k-1}$, this last merge increases the diameter by at least 1 and so the k-clustering produced by complete linkage costs at least k, whereas the optimal clustering consisting of the k individual levels costs 1. □

A Lower Bound for Radius-Based Costs

We show that the instance $(V(P_k), d)$ also yields a lower bound of $k/2$ for radius-based costs. This requires some additional work, as we now also have to keep track of the centers that induce an optimal radius. For an ℓ-graph G_ℓ we again slightly abuse the notation and talk about G_ℓ as a cluster with $\mathsf{drad}(G_\ell) = \min_{c \in V(G_\ell)} \max_{x \in V(G_\ell)} d(c, x)$, the radius of $V(G_\ell)$.

Lemma 3.2.25. *Let $G_{2\ell}$ be any of the 2ℓ-graphs that constitute P_k for $1 \leq \ell \leq \frac{k}{2}$ arbitrary. Then it holds that $\mathsf{drad}(G_{2\ell}) = \ell$ and furthermore, all optimal centers that induce this radius are themselves already contained in $G_{2\ell}$ (and not in any other 2ℓ-graph).*

To prove Lemma 3.2.25 we show that there is a point in P_k for which the following holds:

- For all but one of the ℓ-graphs that constitute $G_{2\ell}$ we can find a point that we can reach by an edge of weight 1. Since the diameter of these graphs is $\ell - 1$, this is sufficient.
- The remaining ℓ-graph lies in the same $(\ell + 1)$-graph as our point and so we are again done by considering the diameter. Also there are no points that induce a smaller radius, since the diameter of $G_{2\ell}$ is already $2\ell - 1$.

Proof of Lemma 3.2.25. By Lemma 3.2.23 we know that the diameter of $G_{2\ell}$ is $2\ell - 1$. Thus the radius of $G_{2\ell}$ is at least ℓ. To show the upper bound of ℓ suppose that $G_{2\ell}$ covers the levels λ up to $\lambda + 2\ell - 1$ in P_k. Consider the unique $(\ell + 1)$-graph $H_{\ell+1}$ contained in $G_{2\ell}$ covering the levels $\lambda + \ell - 1$ to $\lambda + 2\ell - 1$. Let c be the unique point in $H_{\ell+1}$ with level $\lambda + \ell - 1$. By Lemma 3.2.23 the diameter of $H_{\ell+1}$ is ℓ, so any point in $H_{\ell+1}$ is at distance $\leq \ell$ to c. Consider now a point $x \in V(G_{2\ell}) \backslash V(H_{\ell+1})$ and the ℓ-graph H_ℓ containing x. We claim that H_ℓ contains a point y with level $\lambda + \ell - 1$. If this is not true then H_ℓ covers the levels $\lambda + \ell$ up to $\lambda + 2\ell - 1$ and therefore also contains the unique point in $G_{2\ell}$ with level $\lambda + 2\ell - 1$. This is not possible as the unique point in $G_{2\ell}$ with level $\lambda + 2\ell - 1$ is already contained in $H_{\ell+1}$. So using that the diameter of H_ℓ is $\ell - 1$ and $\phi_k(c) = \phi_k(y)$ we obtain

$$d(c, x) \leq d(c, y) + d(y, x) \leq 1 + (\ell - 1) = \ell.$$

Now we prove that all optimal centers must be contained in $G_{2\ell}$. For all points $c \in V(P_k) \backslash V(G_{2\ell})$ we have to show that $\max_{x' \in V(G_{2\ell})} d(c, x') \geq \ell + 1$. Suppose that $\phi_k(c) \leq \lambda + \ell - 1$. Let x be the unique point in $G_{2\ell}$ with level $\lambda + 2\ell - 1$, we claim that $d(c, x) \geq \ell + 1$. Consider a shortest path between c and x and let $\{u, w\}$ be an edge on this path with $u \in V(P_k) \backslash V(G_{2\ell})$ and $w \in V(G_{2\ell})$. By construction $\{u, w\}$ either weights at least 2ℓ in which case

$$d(c, x) \geq 2\ell \geq \ell + 1$$

or it weights 1 and $\phi_k(u) = \phi_k(w)$, so

$$\begin{aligned} d(c, x) &= d(c, u) + d(u, v) + d(v, x) \\ &\geq |\phi_k(c) - \phi_k(u)| + 1 + |\phi_k(w) - \phi_k(x)| \\ &= |\phi_k(c) - \phi_k(x)| + 1 \\ &\geq \ell + 1. \end{aligned}$$

In case $\phi_k(c) \geq \lambda + \ell$ we can prove analogously that $d(c, y) \geq \ell + 1$ for the unique point y in $G_{2\ell}$ with level λ. This finishes the proof. □

Now we make sure that complete linkage completely reconstructs these components. In particular we show that merging 2ℓ-graphs which cover the same levels increases the radius of our solution. Here we make use of the fact that sets of optimal centers for any pair of 2ℓ-graphs do not intersect. Lemma 3.2.26 ensures that the radius indeed increases.

Lemma 3.2.26. *Let C, D be two subsets of $V(P_k)$ with $\mathsf{drad}(C) = \mathsf{drad}(D)$. Let $Z(C)$ and $Z(D)$ denote the set of all optimal centers for C respectively D. If $Z(C) \cap Z(D) = \emptyset$ then $\mathsf{drad}(C \cup D) > \mathsf{drad}(C)$.*

Proof. Let $x \in V(P_k)$. Since $Z(C) \cap Z(D) = \emptyset$ this point can be an optimal center for at most one of the sets. Assume without loss of generality that $x \notin Z(D)$. We have

$$\max_{y \in C \cup D} d(y, x) \geq \max_{y \in D} d(y, x) > \mathsf{drad}(D) = \mathsf{drad}(C)$$

So we have for all $x \in V(P_k)$ that $\max_{y \in C \cup D} d(y, x) > \mathsf{drad}(C)$ which proves the lemma. □

Now, with this we can prove that the merging behavior of complete linkage reconstructs our components. Observe that Theorem 3.2.21 is an immediate consequence of Corollary 3.2.27.

Corollary 3.2.27. *Complete linkage might merge clusters in $(V(P_k), d)$ in such a way that for $1 \leq \ell \leq \frac{k}{2}$, the clustering $\mathcal{C}_\ell$ consists exactly of the 2ℓ-graphs that make up P_k.*

Proof. The proof is an analogous induction to Lemma 3.2.24. Consider the case $\ell = 1$. The first merge increases the radius to 1. Observe by Lemma 3.2.25 that the radius of a 2-graph is 1. Furthermore, the same lemma shows that the sets of optimal centers for any pair of 2-graphs do not intersect and so, as shown in Lemma 3.2.26 any further merge necessarily has to increase the radius. Hence $\mathcal{C}_1$ consists exactly of the 2-graphs.

Assume now that the claim holds for $\mathcal{C}_\ell$. The induction step works essentially the same as the base case. Any merge will increase the radius of the solution by at least 1 by definition of $\mathcal{C}_\ell$ and so we might as well merge all 2ℓ-graphs that together compose a $(2\ell + 2)$-graph as this is a cheapest choice (Lemma 3.2.25). Furthermore, any additional merge would increase the radius to at least $\ell + 2$ (again by Lemma 3.2.26) and so $\mathcal{C}_{\ell+1}$ consists of the $(2\ell + 2)$-graphs. □

Lower Bound for Complete Linkage without Bad Ties

Notice that in our analysis we decided which clusters will be merged by complete linkage whenever it has to choose between two merges of the same cost. However with some adjustments on the instance $(V(P_k), d)$ we can show a lower bound of $\Omega(k)$ for both diameter and radius, for any behavior of complete linkage on ties, as we see next.

We now modify the instance $(V(P_k), d)$ such that merging two ℓ-graphs G_ℓ, G'_ℓ that are part of the same $(\ell + 1)$-graph is slightly cheaper than performing any other merge in a clustering consisting of all ℓ-graphs.

Diameter-Based Cost

We explain how to adjust the construction of the k-components for the diameter. Let $\epsilon \in (0, \frac{1}{2})$. The definition of K_1 stays the same. As before a k-component is constructed from two copies $K_{k-1}^{(0)}, K_{k-1}^{(1)}$ of the $(k-1)$-component by taking the disjoint union of the corresponding graphs and increasing the level of each point in $K_{k-1}^{(1)}$ by one. Here we do not add an edge of weight $k-1$ between the unique points $s \in V(G_{k-1}^{(0)})$ with level 1 and $t \in V(G_{k-1}^{(1)})$ with level k. Instead we complete G_k by adding edges of weight $(k-1)(1-\epsilon)$ between $x \in V(G_{k-1}^{(0)})$ and $y \in V(G_{k-1}^{(1)})$ if they are not on the same level, i.e., $\phi_k(x) \neq \phi_k(y)$.

The instance $(V(P_k), d)$ is then constructed from k copies $K_k^{(1)}, \ldots, K_k^{(k)}$ of the k-component K_k. We take the disjoint union of the corresponding k-graphs $G_k^{(1)}, \ldots, G_k^{(k)}$ and connect them by adding edges $\{x, y\}$ of weight 1 for every two points $x \in V(G_k^{(i)})$ and $y \in V(G_k^{(j)})$ with $\phi_k^{(i)}(x) = \phi_k^{(j)}(y)$.

We show that the clustering computed by complete linkage on $(V(P_k), d)$ at time $t_{\leq \ell(1-\epsilon)}$ consists exactly of the $(\ell+1)$-graphs that make up the instance.

Lemma 3.2.28. *The distance between two points $x, y \in V(P_k)$ is at least $|\phi_k(x) - \phi_k(y)|(1-\epsilon)$.*

Proof. By the inductive construction of the components, an edge which crosses w levels costs at least $w(1-\epsilon)$. Hence the distance between x and y is at least $|\phi_k(x) - \phi_k(y)|(1-\epsilon)$. □

As before we use the previous lemma to show that the diameter of any ℓ-graph in P_k is $(\ell-1)(1-\epsilon)$.

Lemma 3.2.29. *Let G_ℓ be an ℓ-graph contained in P_k. We have $\mathsf{diam}(G_\ell) = (\ell-1)(1-\epsilon)$.*

Proof. We prove the upper bound $\mathsf{diam}(G_\ell) \leq (\ell-1)(1-\epsilon)$ by induction. The 1-graphs are points and so the claim follows trivially for $\ell = 1$. Assume now that we have shown the claim for $\ell - 1$. Let G_ℓ be an ℓ-graph and $s, t \in V(G_\ell)$ points such that $\mathsf{diam}(G_\ell) = d(s,t)$. If these points lie in the same graph, say $G_{\ell-1}^{(0)}$, of the two $(\ell-1)$-graphs $G_{\ell-1}^{(0)}$ and $G_{\ell-1}^{(1)}$ that make up G_ℓ, then

$$d(s,t) \leq \mathsf{diam}(G_{\ell-1}^{(0)}) \leq (\ell-2)(1-\epsilon) < (\ell-1)(1-\epsilon)$$

by induction and we are done. Otherwise we may assume that $s \in V(G_{\ell-1}^{(0)})$ and $t \in V(G_{\ell-1}^{(1)})$. We distinguish two cases. If $\phi_k(s) = \phi_k(t)$ these points are connected by an edge of weight one by construction. Notice that an ℓ-graph does not contain points with the same level if $\ell \leq 2$. Using $\epsilon \leq \frac{1}{2}$ and $\ell \geq 3$ we obtain

$$d(s,t) = 1 \leq (\ell-1)(1-\epsilon).$$

If s and t are on different levels there is an edge of weight $(\ell-1)(1-\epsilon)$ between s and t by construction. Thus we obtain in all cases

$$\mathsf{diam}(G_\ell) = d(s,t) \leq (\ell-1)(1-\epsilon).$$

To see the lower bound $\mathsf{diam}(G_\ell) \geq (\ell-1)(1-\epsilon)$, we apply Lemma 3.2.28 to the unique point $s \in V(G_\ell)$ with $\phi_\ell(s) = 1$ and the unique point $t \in V(G_\ell)$ with $\phi_\ell(t) = \ell$. This shows that $\mathsf{diam}(G_\ell) \geq d(s,t) \geq (\ell-1)(1-\epsilon)$. □

We show that complete linkage must reconstruct these components as clusters.

Lemma 3.2.30. *Complete linkage must merge clusters on $(V(P_k), d)$ in such a way that for all $\ell < k$, the clustering $\mathcal{C}_{\ell(1-\epsilon)}$ consists exactly of the $(\ell+1)$-graphs that make up P_k.*

Proof. We prove the claim by induction. Complete linkage always starts with every point in a separate cluster. Since those are exactly the 1-graphs and any merge of two points costs at least $(1-\epsilon)$, the claim follows for $\ell = 0$. Suppose now that $\mathcal{C}_{(\ell-1)(1-\epsilon)}$ consists exactly of the ℓ-graphs of the instance. Consider two ℓ-graphs $G_\ell \neq G'_\ell$ contained in the current clustering. We compute the cost of merging G_ℓ with G'_ℓ. For this purpose we distinguish whether they are contained in the same $(\ell+1)$-graph or not.

Case 1: If they are contained in the same $(\ell+1)$-graph $G_{\ell+1}$ merging G_ℓ with G'_ℓ results in $G_{\ell+1}$. We obtain by Lemma 3.2.29 that $\mathsf{diam}(G_{\ell+1}) = \ell(1-\epsilon)$.

Case 2: If they are not contained in the same $(\ell+1)$-graph, we show that merging G_ℓ with G'_ℓ costs more than $\ell(1-\epsilon)$. We make the following observations.

1. The edges connecting $x \in V(G_\ell)$ and $y \in V(G'_\ell)$ with $\phi_k(x) \neq \phi_k(y)$ are of weight $\geq (\ell+1)(1-\epsilon)$.
2. There exist $s \in V(G_\ell)$ and $t \in V(G'_\ell)$ with $|\phi_k(s) - \phi_k(t)| \geq \ell - 1$.

The last observation follows from the fact that each of the graphs contains two points whose difference in levels is exactly $\ell - 1$.

We prove that $d(s,t) > \ell(1-\epsilon)$ and therefore merging G_ℓ with G'_ℓ costs more than $\ell(1-\epsilon)$. Any shortest path connecting s and t in P_k must contain an edge $\{u,w\}$ between a point $u \in V(G_\ell)$ and a point $w \in V(G'_\ell)$. By above observation this edge is either of weight $\geq (\ell+1)(1-\epsilon)$ or u and w are on the same level and the edge is of weight 1. In the first case we conclude

$$d(s,t) \geq (\ell+1)(1-\epsilon) > \ell(1-\epsilon).$$

In the second case we obtain that

$$\begin{aligned} d(s,t) &= d(s,u) + 1 + d(w,t) \\ &\geq |\phi_k(s) - \phi_k(u)|(1-\epsilon) + 1 + |\phi_k(w) - \phi_k(t)|(1-\epsilon) \\ &= |\phi_k(s) - \phi_k(t)|(1-\epsilon) + 1 \\ &\geq (\ell-1)(1-\epsilon) + 1 \\ &> \ell(1-\epsilon). \end{aligned}$$

We see that $\mathcal{C}_{\ell(1-\epsilon)}$ must consist exactly of the $(\ell+1)$-graphs of P_k. □

Lemma 3.2.30 shows that $\mathcal{C}_{(k-1)(1-\epsilon)}$ consists of all the k-graphs that make up P_k. There are exactly k of them, thus the k-clustering produced by complete linkage costs $(k-1)(1-\epsilon)$.

Corollary 3.2.31. *However the tie-breaks are resolved, complete linkage computes a k-clustering on $(V(P_k), d)$ with diameter $(k-1)(1-\epsilon)$ while the optimal k-clustering has diameter 1.*

Radius-Based Cost

We explain how to adjust the construction of the k-components for the radius. Let $\epsilon \in (0, \frac{1}{2})$. The definition of K_1 does not change. As before a k-component is constructed from two copies $K_{k-1}^{(0)}, K_{k-1}^{(1)}$ of the $(k-1)$-component by taking the disjoint union of the corresponding graphs and increasing the level of each point in $K_{k-1}^{(1)}$ by one. We complete G_k by adding edges between $x \in V(G_{k-1}^{(0)})$ and $y \in V(G_{k-1}^{(1)})$ if $\phi_k(x) \neq \phi_k(y)$ and we assign this edge a weight of $\lceil \frac{k}{2} \rceil (1-\epsilon)$ if $|\phi_k(x) - \phi_k(y)| \leq \lceil \frac{k}{2} \rceil - 1$ and otherwise a weight of $|\phi_k(x) - \phi_k(y)|(1-\epsilon)$.

As before the instance $(V(P_k), d)$ is constructed from k copies $K_k^{(1)}, \ldots, K_k^{(k)}$ of the k-component K_k. We take the disjoint union of the corresponding k-graphs $G_k^{(1)}, \ldots, G_k^{(k)}$ and connect them by adding edges $\{x, y\}$ of weight 1 for every two points $x \in V(G_k^{(i)})$ and $y \in V(G_k^{(j)})$ with $\phi_k^{(i)}(x) = \phi_k^{(j)}(y)$. We observe that Lemma 3.2.28 still holds on the adjusted instance. Also notice that the diameter of an ℓ-graph is still upper bounded by $(\ell - 1)(1-\epsilon)$.

Lemma 3.2.32. *Let $G_{2\ell}$ be any of the 2ℓ-graphs that constitute P_k for $1 \leq \ell \leq \frac{k}{2}$. It holds that $\mathsf{drad}(G_{2\ell}) = \ell(1-\epsilon)$. Furthermore let $G'_{2\ell}$ be a second 2ℓ-graph which is not contained in the same $2(\ell+1)$-graph as $G_{2\ell}$. Any cluster containing $G_{2\ell}$ and $G'_{2\ell}$ costs at least $\ell(1-\epsilon)+1$.*

Proof. We know that $G_{2\ell}$ contains points s and t with $|\phi_k(s) - \phi_k(t)| = 2\ell - 1$. Thus for any $x \in V(P_k)$ we have $\max\{|\phi_k(s) - \phi_k(x)|, |\phi_k(t) - \phi_k(x)|\} \geq \ell$. By Lemma 3.2.28 we know that $\max\{d(s,x), d(t,x)\} \geq \ell(1-\epsilon)$ and therefore $\mathsf{drad}(G_{2\ell}) \geq \ell(1-\epsilon)$.

To prove the upper bound suppose that $G_{2\ell}$ covers the levels λ up to $\lambda + 2\ell - 1$ in P_k. Consider the unique $(\ell+1)$-graph $H_{\ell+1}$ contained in $G_{2\ell}$ covering the levels $\lambda + \ell - 1$ to $\lambda + 2\ell - 1$. Let c be the unique point in $H_{\ell+1}$ with level $\lambda + \ell - 1$. Remember that the diameter of $H_{\ell+1}$ is at most $\ell(1-\epsilon)$, so any point in $H_{\ell+1}$ is at distance $\leq \ell(1-\epsilon)$ to c. Consider now a point $x \in V(G_{2\ell}) \backslash V(H_{\ell+1})$. We know that $\phi_k(x) < \lambda + 2\ell - 1$. Thus $|\phi_k(x) - \phi_k(c)| \leq \ell - 1$. By construction there exists an edge of weight at most $\ell(1-\epsilon)$ between x and c and thus $d(x,c) \leq \ell(1-\epsilon)$.

It is left to show that any cluster containing $G_{2\ell}$ and $G'_{2\ell}$ costs at least $\ell(1-\epsilon)+1$. Let $y \in V(P_k)$ and let $H_{2(\ell+1)}$ be the $2(\ell+1)$-graph containing y. Assume without loss of generality that $G_{2\ell}$ is not contained in $H_{2(\ell+1)}$. Let $x \in V(G_{2\ell})$ be a point with $|\phi_k(x) - \phi_k(y)| \geq \ell$. We claim that $d(x,y) \geq (\ell-1)(1-\epsilon)+1$. A shortest path connecting x and y must contain an edge $\{u, w\}$ with $u \in V(P_k) \backslash V(H_{2(\ell+1)})$ and $w \in V(H_{2(\ell+1)})$. We know by construction that either $\phi_k(u) = \phi_k(w)$, or the edge weights at least $(\ell+2)(1-\epsilon)$. In the first case we use Lemma 3.2.28 and obtain

$$\begin{aligned} d(x,y) &= d(x,u) + d(u,w) + d(w,y) \\ &\geq |\phi_k(x) - \phi_k(u)|(1-\epsilon) + 1 + |\phi_k(w) - \phi_k(y)|(1-\epsilon) \\ &= |\phi_k(x) - \phi_k(y)|(1-\epsilon) + 1 \\ &\geq \ell(1-\epsilon) + 1 \end{aligned}$$

and in the second case we obtain

$$d(x,y) \geq (\ell+2)(1-\epsilon) \geq \ell(1-\epsilon) + 1. \qquad \square$$

This immediately leads to the following results.

Corollary 3.2.33. *Complete linkage must merge clusters on $(V(P_k), d)$ in such a way that for all $1 \leq \ell \leq \frac{k}{2}$, the clustering $\mathcal{C}_{\ell(1-\epsilon)}$ consists exactly of the 2ℓ-graphs that make up P_k.*

Corollary 3.2.34. *However the tie-breaks are resolved, complete linkage computes a k-clustering on $(V(P_k), d)$ with radius $\frac{k}{2}(1-\epsilon)$, while the optimal k-clustering has radius 1.*

3.2.4 The Average Approximation Factor

We have seen previously that the approximation guarantee for complete linkage for the radius is in $\Theta(k)$ and that the same holds for single linkage. This is rather surprising since complete linkage merges clusters based on which merge minimizes the objective function, which is not the case for single linkage. Notice that when we perform complete linkage on the worst case instance $(V(P_k), d)$ presented in Section 3.2.3 complete linkage produces reasonably good clusters for most cluster sizes $\ell \neq k$. Depending on the application we may not need a strong approximation guarantee for all cluster sizes, instead it may be sufficient to find a hierarchical clustering which is a good approximation to the optimal cost for most of the cluster sizes. We try to incorporate this by considering the average approximation factor. The advantage of this new definition is emphasized by the fact that complete linkage for drad performs asymptotically better than single linkage with respect to this definition. This fits our intuition that complete linkage is more suited for the task of constructing hierarchical clusterings for the drad objective than single linkage even though both compute a $\Theta(k)$-approximation with respect to the standard definition of an approximation factor.

Definition 3.2.35. *Let $(\mathcal{H}_k)_{k=1}^n$ be an arbitrary hierarchical clustering on $(\mathcal{P}, d)$ and let $(\mathcal{O}_k)_{k=1}^n$ be optimal solutions for the radius or diameter. We denote by*

$$\mathsf{Avg}((\mathcal{H}_k)_{k=1}^n) = \frac{1}{n}\sum_{k=1}^{n} \frac{\mathsf{cost}(\mathcal{H}_k)}{\mathsf{cost}(\mathcal{O}_k)}$$

the average approximation factor of $(\mathcal{H}_k)_{k=1}^n$.

The following corollary is an immediate consequence of Lemma 3.2.7.

Corollary 3.2.36. *Let $(\mathcal{H}_k)_{k=1}^n$ be the hierarchical clustering computed by complete linkage for the radius. We have*

$$\mathsf{Avg}((\mathcal{H}_k)_{k=1}^n) \leq \lceil \log_2(n) \rceil.$$

However the upper bound of $\lceil \log_2(n) \rceil$ seems too pessimistic. It would be interesting to know whether this bound is tight or complete linkage in fact computes a constant factor approximation on average and whether similar results hold for the diameter.

Next we give a lower bound on the average approximation factor for single linkage. Let $n = 2^s$. Consider the instance $\mathcal{P} = \{1, \ldots, n\} \subset \mathbb{R}$. We can assume that in the k-th step single linkage merges the two clusters containing x_{n-k} and x_{n-k+1} as the distance between these clusters is 1. The k-clustering computed by single linkage on $(\mathcal{P}, \|\cdot\|_1)$ then equals

$$\mathcal{H}_k = \{\{x_1\}, \ldots, \{x_{k-1}\}, \{x_k, \ldots, x_n\}\}$$

and has diameter $n - k$.

On the other hand for $0 \leq t \leq s$ the optimal 2^t-clustering has diameter $2^{s-t}-1$ and consists of clusters with 2^{s-t} consecutive points in $\mathcal{P}$:

$$\mathcal{O}_{2^t} = \{\{x_1, \ldots, x_{2^{s-t}}\}, \ldots, \{x_{2^{s-t}(2^t-1)+1}, \ldots, x_{2^s}\}\}$$

Thus we obtain for the diameter

$$\begin{aligned}
\mathsf{Avg}((\mathcal{H}_k)_{k=1}^n) &= \frac{1}{n}\Big(1 + \sum_{t=0}^{s-1} \sum_{k \in (2^t, 2^{t+1}]} \frac{\mathsf{diam}(\mathcal{H}_k)}{\mathsf{diam}(\mathcal{O}_k)}\Big) \geq \frac{1}{n}\sum_{t=0}^{s-1} 2^t \frac{\mathsf{diam}(\mathcal{H}_{2^{t+1}})}{\mathsf{diam}(\mathcal{O}_{2^t})} \\
&\geq \frac{1}{n}\sum_{t=0}^{s-1} 2^t \frac{2^s - 2^{t+1}}{2^{s-t}} = \frac{1}{n}\sum_{t=0}^{s-1} 2^{2t} \frac{2^{s-t}-2}{2^{s-t}} \\
&= \frac{1}{n}\Big(\sum_{t=0}^{s-1} 4^t - \sum_{t=0}^{s-1} \frac{2^{3t}}{2^{s-1}}\Big) = \frac{1}{2^s}\Big(\sum_{t=0}^{s-1} 4^t - \frac{1}{2^{s-1}}\sum_{t=0}^{s-1} 8^t\Big) \\
&= \frac{1}{2^s}\frac{4^s-1}{3} - \frac{1}{2^{2s-1}}\frac{8^s-1}{7} = \frac{2^s - 2^{-s}}{3} - \frac{2^{s+1} - 2^{1-2s}}{7} \\
&\geq \frac{n}{21} - 1
\end{aligned}$$

The same computation can be done for the radius, as the radius of $\mathcal{H}_{2^{t+1}}$ equals $\frac{2^s - 2^{t+1}}{2}$ and the radius of $\mathcal{O}_{2^t}$ equals $\frac{2^{s-t}}{2}$.

Corollary 3.2.37. *The average approximation factor achieved by single linkage on $(\mathcal{P}, \|\cdot\|_1)$ for both radius and diameter is at least $\frac{n}{21} - 1$.*

We conclude that the average approximation factor achieved by complete linkage for drad is asymptotically better than the average approximation factor achieved by single linkage. In general it may be of interest to consider other ways to measure the quality of hierarchical clusterings computed by complete linkage and single linkage.

3.3 Complete Linkage in the Euclidean Space

We have seen that the approximation guarantee of complete linkage in general metric spaces is in $\Omega(k)$ for the radius and diameter objective. Remember that for the analysis of the radius for a fixed cluster size k we made use of the fact that when a clustering $\mathcal{H}_{k'}$ is computed by complete linkage for $k' > k$ the next $\frac{k'-k}{2}$ merges performed by complete linkage increase the radius by at most $2\mathsf{drad}(\mathcal{O})$ where $\mathcal{O}$ is the optimal k-clustering with respect to drad. This bound on the increase in cost can be improved if we assume that the given metric space is Euclidean. Ackermann et al. [1] use this fact to give better approximation bounds in the Euclidean space. From now on let $\|\cdot\|_2$ denote the Euclidean metric.

Definition 3.3.1 ([1])**.** *Given a set of points $\mathcal{P} \subset \mathbb{R}^d$ and $k \in \mathbb{N}, r \in \mathbb{R}$. Then $\mathcal{P}$ is called (k,r)-coverable, if there exist k points $c_1, \ldots, c_k$ such that $\mathcal{P} \subset \bigcup_{i=1}^k B(c_i, r)$, where $B(c_i, r) = \{x \in \mathbb{R}^d \mid \|x - c_i\|_2 \leq r\}$ is the ball of radius r around c_i.*

Lemma 3.3.2 ([1])**.** *Given some $k \in \mathbb{N}, r \in \mathbb{R}$ and a set of more than k points $\mathcal{P} \subset \mathbb{R}^d$ which is (k,r)-coverable, there exist two distinct points $x, y \in \mathcal{P}$ with $\|x - y\|_2 \leq 4r\sqrt[d]{\frac{k}{|\mathcal{P}|}}$.*

Using Lemma 3.3.2 Ackermann et al. are able to proof that complete linkage produces a clustering with few clusters whose cost is not too large when compared to the optimal k-clustering $\mathcal{O}_k$.

Definition 3.3.3. *Given the clustering instance* $(\mathcal{X}, \mathcal{P}, d)$ *and* $\mathsf{cost} \in \{\mathsf{drad}, \mathsf{rad}, \mathsf{diam}\}$. *Let* $\mathcal{C}$ *be an arbitrary clustering of* $\mathcal{P}$ *and* $\mathcal{O}_k$ *the optimal clustering of size* k *with respect to* cost. *For* $1 \leq k \leq n$ *we consider the graph* $G[\mathcal{C}, \mathcal{O}_k]$ *with vertex set* $\mathcal{O}_k$, *where we connect two vertices* $O, O' \in \mathcal{O}_k$ *by an edge if there is a cluster* $C \in \mathcal{C}$ *which intersects both* O *and* O'.

Furthermore we say that $\mathcal{C}$ *is* 2*-near to* $\mathcal{O}_k$ *if for every connected component* $G = (V, E)$ *of* $G[\mathcal{C}, \mathcal{O}_k]$ *the number of clusters in* $\mathcal{C}$ *which intersect at least one vertex in* V *is bounded by* $2|V|$.

Lemma 3.3.4 ([1])**.** *Given the clustering instance* $(\mathbb{R}^d, \mathcal{P}, \|\cdot\|_2)$ *and* $\mathsf{cost} \in \{\mathsf{drad}, \mathsf{rad}, \mathsf{diam}\}$, *with respect to* cost *complete linkage computes a clustering* $\mathcal{H}_{k'}$ *that is* 2*-near to* $\mathcal{O}_k$ *and whose cost can be upper bounded by*

1. $10d\mathsf{drad}(\mathcal{O}_k)$ *if* $\mathsf{cost} = \mathsf{drad}$
2. $24de^{24d}\mathsf{rad}(\mathcal{O}_k)$ *if* $\mathsf{cost} = \mathsf{rad}$
3. $2^{3\cdot(42d)^d}(28d+6)\mathsf{diam}(\mathcal{O}_k)$ *if* $\mathsf{cost} = \mathsf{diam}$

where $\mathcal{O}_k$ *denotes the optimal* k*-clustering with respect to* cost.

So we see that under the assumption that the dimension of the Euclidean space is constant, there exists a clustering with few clusters computed by complete linkage whose cost can be upper bounded by $O(1)\mathsf{cost}(\mathcal{O}_k)$. Ackermann et al. then show that the remaining merge steps performed by complete linkage to obtain the clustering $\mathcal{H}_k$ increase the cost of the solution by a factor of at most $O(\log(k))$. The analysis of the remaining merge steps was later improved by Großwendt and Röglin [44]. They showed the following statement, which also holds in general metric spaces.

Theorem 3.3.5 ([44])**.** *Given the clustering instance* $(\mathcal{X}, \mathcal{P}, d)$ *and* $\mathsf{cost} \in \{\mathsf{drad}, \mathsf{rad}, \mathsf{diam}\}$, *let* $\mathcal{H}_{k'}$ *be a* 2*-near clustering to* $\mathcal{O}_k$ *and* $\mathcal{H}_k$ *be the* k*-clustering computed by complete linkage with respect to* cost. *We have that*

1. $\mathsf{cost}(\mathcal{H}_k) \leq 25\mathsf{cost}(\mathcal{H}_{k'}) + 12\mathsf{cost}(\mathcal{O}_k)$ *if* $\mathsf{cost} = \mathsf{drad}$
2. $\mathsf{cost}(\mathcal{H}_k) \leq 13\mathsf{cost}(\mathcal{H}_{k'}) + 6\mathsf{cost}(\mathcal{O}_k)$ *if* $\mathsf{cost} = \mathsf{rad}$
3. $\mathsf{cost}(\mathcal{H}_k) \leq 9\mathsf{cost}(\mathcal{H}_{k'}) + 8\mathsf{cost}(\mathcal{O}_k)$ *if* $\mathsf{cost} = \mathsf{diam}$

Combining Lemma 3.3.4 amd Theorem 3.3.5 yields immediately the following corollary.

Corollary 3.3.6. *Given the clustering instance* $(\mathbb{R}^d, \mathcal{P}, \|\cdot\|_2)$ *and* $\mathsf{cost} \in \{\mathsf{drad}, \mathsf{rad}, \mathsf{diam}\}$, *with respect to* cost *complete linkage computes a clustering* $\mathcal{H}_k$ *of size* k *whose cost can be upper bounded by*

1. $(250d+12)\mathsf{drad}(\mathcal{O}_k)$ *if* $\mathsf{cost} = \mathsf{drad}$
2. $(312de^{24d}+6)\mathsf{rad}(\mathcal{O}_k)$ *if* $\mathsf{cost} = \mathsf{rad}$

3. $(9 \cdot 2^{3\cdot(42d)^d}(28d+6)+8)\mathsf{diam}(\mathcal{O}_k)$ *if* $\mathsf{cost} = \mathsf{diam}$

where $\mathcal{O}_k$ *denotes the optimal* k*-clustering with respect to* cost*.*

In this section we now give a simplified proof of the analysis presented in [44], which also yields slightly better approximation factors for all three objectives.

We start by observing a simple property about trees.

Lemma 3.3.7. *Let* $T = (V, E)$ *be a tree with* $m > 1$ *vertices. There exists a set* M *of at most* $\frac{m}{2}$ *vertices such that every other vertex in* V *is incident to a vertex in* M*.*

Proof. We prove this by induction on m. For $m = 2, 3$ the claim holds. If $m > 3$ suppose T is rooted at a vertex v and let h be the height of T with respect to v. Let $w \in V$ be a vertex of height $h-1$ that is not a leaf and denote with Z the children of w. Observe that $T' = T \backslash (Z \cup \{w\})$ is still connected, as all children of w are of height h and therefore must be leaves. If T' is empty or consists of only v, then the lemma holds for $M = \{w\}$. In the other case T' consists of $m' \geq 2$ vertices. Thus we can apply the induction hypothesis on T' to obtain a set M' of at most $\frac{m'}{2}$ vertices. We set $M = M' \cup \{w\}$ and observe that all vertices in T are incident to at least one vertex in M. Furthermore $|M| \leq \frac{m'}{2} + 1 = \frac{m'+2}{2}$. Now T' contains at least two vertices fewer than T as $Z \neq \emptyset$ and therefore $m' + 2 \leq m$. □

Lemma 3.3.8. *Let* $\mathcal{C}$ *be a clustering of* $\mathcal{P}$ *and* $\mathcal{O}_k$ *the optimal* k*-clustering with respect to* $\mathsf{cost} \in \{\mathsf{drad}, \mathsf{rad}, \mathsf{diam}\}$*. Let* $G = (V, E)$ *be a connected component of* $G[\mathcal{C}, \mathcal{O}_k]$ *with* $m > 1$ *vertices. There exists a clustering of cost at most* $3\mathsf{cost}(\mathcal{O}_k) + 2\mathsf{cost}(\mathcal{C})$ *for* $\mathsf{cost} \in \{\mathsf{drad}, \mathsf{rad}, \mathsf{diam}\}$*, which is obtained by merging at least* $\frac{m}{2}$ *clusters from* V*.*

Proof. We delete edges from G until we obtain a tree. By Lemma 3.3.7 there exists a set M of at most $\frac{m}{2}$ vertices such that every other vertex in G is incident to a vertex from M. We reduce the number of clusters by merging every cluster $O \in V \backslash M$ with a cluster in M it is incident to. The resulting clustering consists of at least $\frac{m}{2}$ fewer cluster than $\mathcal{O}_k$. Let C be one of the merged clusters and let $O \in M$ be the unique cluster from M contained in C. We upper bound the cost of C in terms of $\mathsf{cost}(\mathcal{O}_k)$ and $\mathsf{cost}(\mathcal{C})$.

When we are dealing with the radius and discrete radius, i.e., $\mathsf{cost} \in \{\mathsf{drad}, \mathsf{rad}\}$ let o be the center of O. We know that every point $p \in C$ is either contained in O itself, in which case $d(p, o) \leq \mathsf{cost}(\mathcal{O}_k)$, or in a cluster $O' \in V \backslash M$. As O and O' are connected by an edge, there must be a cluster in $\mathcal{C}$ which intersects both. Let $q_1 \in O'$ and $q_2 \in O$ be two points contained in the same cluster of $\mathcal{C}$. We obtain

$$\begin{aligned} d(p, o) &\leq d(p, q_1) + d(q_1, q_2) + d(q_2, o) \\ &\leq 3\mathsf{cost}(\mathcal{O}_k) + 2\mathsf{cost}(\mathcal{C}). \end{aligned}$$

When we are dealing with the diameter $\mathsf{cost} = \mathsf{diam}$ we make a similar computation. Let $p_1, p_2 \in C$ be contained in the optimal clusters O_1 or O_2, respectively. We know that there are $q_1 \in O_1, q_2 \in O$ and $r_1 \in O, r_2 \in O_2$ such that for each pair both points lie in the same cluster in $\mathcal{C}$. We obtain

$$\begin{aligned} d(p_1, p_2) &\leq d(p_1, q_1) + d(q_1, q_2) + d(q_2, r_1) + d(r_1, r_2) + d(r_2, p_2) \\ &\leq 3\mathsf{cost}(\mathcal{O}_k) + 2\mathsf{cost}(\mathcal{C}). \end{aligned}$$

□

Thus we showed that we can roughly half the number of clusters by merging clusters in $\mathcal{O}_k$ such that the resulting clustering $\mathcal{O}'$ is not too expensive. Let $\mathscr{H} = (\mathcal{H}_{|\mathcal{P}|}, \dots \mathcal{H}_1)$ denote the hierarchical clustering that is computed by complete linkage. Next we fix a complete linkage clustering $\mathcal{H}$ with more than k clusters and use $\mathcal{O}_k$ or $\mathcal{O}'$ to argue that complete linkage computes a clustering with significantly fewer clusters than $\mathcal{H}$ while the increase in cost can be bounded in terms of $\mathsf{cost}(\mathcal{O}_k)$ or $\mathsf{cost}(\mathcal{O}')$.

In our analysis we again consider the smallest clustering from $\mathscr{H}$ whose cost does not exceed a given bound x, which we denote by $\mathcal{C}_x$. Remember that every merge of two clusters in $\mathcal{C}_x$ results in a clustering of cost more than x. We show the following improved version of Theorem 3.3.5 by Großwendt and Röglin.

Lemma 3.3.9. *Given the clustering instance $(\mathcal{X}, \mathcal{P}, d)$ and let $\mathsf{cost} \in \{\mathsf{drad}, \mathsf{rad}, \mathsf{diam}\}$ and let $\mathcal{H}$ be a clustering created by complete linkage with $\mathsf{cost}(\mathcal{H}) > 2\mathsf{cost}(\mathcal{O}_k)$ and which is 2-near to $\mathcal{O}_k$, then we have*

1. $\mathsf{cost}(\mathcal{H}_k) \leq 8\mathsf{cost}(\mathcal{O}_k) + 5\mathsf{cost}(\mathcal{H})$ *for* $\mathsf{cost} = \mathsf{drad}$
2. $\mathsf{cost}(\mathcal{H}_k) \leq 5\mathsf{cost}(\mathcal{O}_k) + 6\mathsf{cost}(\mathcal{H})$ *for* $\mathsf{cost} = \mathsf{rad}$
3. $\mathsf{cost}(\mathcal{H}_k) \leq 5\mathsf{cost}(\mathcal{O}_k) + 6\mathsf{cost}(\mathcal{H})$ *for* $\mathsf{cost} = \mathsf{diam}$.

Proof. Let $G = (V, E)$ be the union of all connected components of $G[\mathcal{H}, \mathcal{O}_k]$ consisting of more than one vertex and $\mathcal{D} \subset \mathcal{H}$ be the clusters which intersect at least one cluster in V. Notice that $|\mathcal{D}| \leq 2|V|$ since $\mathcal{H}$ is 2-near to $\mathcal{O}_k$. Let $\ell = |V|$. First observe that $\mathcal{H}\backslash\mathcal{D}$ consists of at most $k - \ell$ clusters. If a cluster $C \in \mathcal{H}\backslash\mathcal{D}$ intersects a cluster $O \in \mathcal{O}_k\backslash V$, then we already have $C \subset O$, as otherwise O would be incident to another optimal cluster in $G[\mathcal{H}, \mathcal{O}_k]$ and therefore contained in a connected component of size at least two. Furthermore there cannot be two clusters $C_1, C_2 \in \mathcal{H}$ contained in O as merging C_1 with C_2 yields a cluster of cost at most $2\mathsf{cost}(\mathcal{O}_k)$. As complete linkage always chooses the merge with smallest increase in cost and $\mathsf{cost}(\mathcal{H}) > 2\mathsf{cost}(\mathcal{O}_k)$ this case does not occur. Thus $|\mathcal{H}\backslash\mathcal{D}| \leq |\mathcal{O}_k\backslash V| = k - \ell$ and so $|\mathcal{H}| = |\mathcal{D}| + |\mathcal{H}\backslash\mathcal{D}| \leq k + \ell$.

Let $x = \mathsf{cost}(\mathcal{O}_k)$ and $y = \mathsf{cost}(\mathcal{H})$. We denote by $\mathcal{O}'$ the clustering of cost at most $z = 3x + 2y$ obtained from $\mathcal{O}_k$ by merging at least $\frac{\ell}{2}$ cluster from V. Such a clustering exists by Lemma 3.3.8.

$\mathsf{cost} = \mathsf{drad}$**:** Let $C, D \subset \mathcal{P}$ be two arbitrary clusters with centers c, d. Observe that

$$\mathsf{cost}(C \cup D) \leq \max\{\mathsf{cost}(C), \mathsf{cost}(D)\} + d(c, d). \tag{3.5}$$

We claim that first $|\mathcal{C}_y \cap \mathcal{C}_{y+2x}| \leq k$ and second $|\mathcal{C}_{y+2x} \cap \mathcal{C}_{y+2x+2z}| \leq k - \frac{\ell}{2}$.

Suppose the first inequality does not hold, thus there must be two clusters C, D in $\mathcal{C}_y \cap \mathcal{C}_{y+2x}$ whose centers are contained in the same cluster from $\mathcal{O}_k$. We use (3.5) and get $\mathsf{cost}(C \cup D) \leq \max\{\mathsf{cost}(C), \mathsf{cost}(D)\} + 2\mathsf{cost}(\mathcal{O}_k) \leq y + 2x$ in contradiction to the definition of $\mathcal{C}_{y+2x}$. Similarly one can prove the second inequality by replacing $\mathcal{O}_k$ by $\mathcal{O}'$. If the second inequality is violated, then there must be two clusters C, D in $\mathcal{C}_{y+2x} \cap \mathcal{C}_{y+2x+2z}$ with $\mathsf{cost}(C \cup D) \leq \max\{\mathsf{cost}(C), \mathsf{cost}(D)\} + 2\mathsf{cost}(\mathcal{O}') \leq y + 2x + 2z$ in contradiction to the definition of $\mathcal{C}_{y+2x+2z}$.

We prove later that $|\mathcal{C}_{y+2x+2z}| \leq k$. Under this assumption we obtain $\mathsf{cost}(\mathcal{H}_k) \leq \mathsf{cost}(\mathcal{C}_{y+2x+2z}) \leq y + 2x + 2z = 8\mathsf{cost}(\mathcal{O}_k) + 5\mathsf{cost}(\mathcal{H})$, which proves the lemma for the discrete radius.

$\mathsf{cost} = \mathsf{rad}$**:** Let $C, D \subset \mathcal{P}$ be two arbitrary clusters. We claim that first $|\mathcal{C}_y \cap \mathcal{C}_{2y+x}| \leq k$ and second $|\mathcal{C}_{2y+x} \cap \mathcal{C}_{4y+2x+z}| \leq k - \frac{\ell}{2}$. If the first inequality is violated we find two

clusters C, D in $\mathcal{C}_y \cap \mathcal{C}_{2y+x}$ such that both intersect the same cluster O from $\mathcal{O}_k$. Let o be the center of that cluster O and $p_1 \in C \cap O, p_2 \in D \cap O$. Observe that $\mathsf{cost}(C \cup D) \leq \max_{p \in C \cup D} d(p, o) \leq 2 \max\{\mathsf{cost}(C), \mathsf{cost}(D)\} + \max\{d(p_1, o), d(p_2, o)\} \leq 2y + x$ which yields a contradiction. Analogously one can prove the second inequality.

We prove later that $|\mathcal{C}_{4y+2x+z}| \leq k$. Under this assumption we obtain $\mathsf{cost}(\mathcal{H}_k) \leq \mathsf{cost}(\mathcal{C}_{4y+2x+z}) \leq 4y + 2x + z = 5\mathsf{cost}(\mathcal{O}_k) + 6\mathsf{cost}(\mathcal{H})$, which proves the lemma for the radius.

$\mathsf{cost} = \mathsf{diam}$**:** Let $C, D \subset \mathcal{P}$ be two arbitrary clusters and $p \in C, q \in D$. Then we have

$$\mathsf{cost}(C \cup D) \leq \mathsf{cost}(C) + d(p, q) + \mathsf{cost}(D). \tag{3.6}$$

We claim that first $|\mathcal{C}_y \cap \mathcal{C}_{2y+x}| \leq k$ and second $|\mathcal{C}_{2y+x} \cap \mathcal{C}_{4y+2x+z}| \leq k - \frac{\ell}{2}$. The argument is again the same. If one of the inequalities is violated we find two clusters C, D in $\mathcal{C}_y \cap \mathcal{C}_{2y+x}$ or $\mathcal{C}_{2y+x} \cap \mathcal{C}_{4y+2x+z}$ such that both intersect the same cluster from $\mathcal{O}_k$ or $\mathcal{O}'$, respectively. By (3.6) $\mathsf{cost}(C \cup D)$ is upper bounded by $2y + x$ or $2(2y + x) + z$ which yields a contradiction.

We prove later that $|\mathcal{C}_{4y+2x+z}| \leq k$. Under this assumption we obtain that $\mathsf{cost}(\mathcal{H}_k) \leq \mathsf{cost}(\mathcal{C}_{4y+2x+z}) \leq 4y + 2x + z = 5\mathsf{cost}(\mathcal{O}_k) + 6\mathsf{cost}(\mathcal{H})$, which proves the lemma for the diameter.

Let $0 \leq a \leq b \leq c$ such that $|\mathcal{C}_a \cap \mathcal{C}_b| \leq k, |\mathcal{C}_b \cap \mathcal{C}_c| \leq k - \frac{\ell}{2}$ and $|\mathcal{C}_a| \leq k + \ell$. We know that all clusters in $\mathcal{C}_a \backslash \mathcal{C}_b$ or $\mathcal{C}_b \backslash \mathcal{C}_c$ must be merged in $\mathcal{C}_b$ or $\mathcal{C}_c$, respectively. Thus we obtain the following bounds on the size of both clusterings.

$$\begin{aligned} |\mathcal{C}_b| &\leq \frac{|\mathcal{C}_a| - k}{2} + k \\ |\mathcal{C}_c| &\leq \frac{|\mathcal{C}_b| - (k - \ell/2)}{2} + k - \frac{\ell}{2} \leq \frac{|\mathcal{C}_a| - k}{4} + k - \frac{\ell}{4} \leq k. \end{aligned}$$

For the last inequality we use $|\mathcal{C}_a| \leq k + \ell$.

We apply this observation for all three objectives and use that $|\mathcal{C}_y| \leq |\mathcal{H}| \leq k + \ell$ to see that indeed $|\mathcal{C}_{y+2x+2z}| \leq k$ for discrete radius and $|\mathcal{C}_{4y+2x+z}| \leq k$ for radius and diameter. □

Chapter 4

Minimum-Error Triangulation Is NP-Hard

This chapter contains results from the work *Minimum-Error Triangulations for Sea Surface Reconstruction* [11] by Anna Arutyunova, Anne Driemel, Jan-Henrik Haunert, Herman Haverkort, Jürgen Kusche, Elmar Langetepe, Philip Mayer, Petra Mutzel and Heiko Röglin published in the proceedings of the *International Symposium on Computational Geometry (SoCG), 2022*. The first part of the paper contains an NP-hardness proof of the minimum-error triangulation problem. The second part considers a dynamic-programming approach to compute minimum error triangulations of the sea surface, where the set of allowed triangulations is restricted to so called k-OD triangulations. In this chapter we present only the NP-hardness proof while the second part of the paper is omitted. A full version of this paper is also available at *arXiv* [12] and currently under consideration at the *Journal of Computational Geometry.*

4.1 The Planar 3SAT Problem

To prove that the minimum-error triangulation problem is NP-hard we perform a reduction from the planar 3SAT problem. The Boolean satisfiability problem (SAT problem) is the first problem that has been proven to be NP-complete [35]. This result is often referred as the Cook-Levin theorem.

Definition 4.1.1 (SAT)**.** *For a set of variables V we define the set of literals as $V \cup \{\overline{v} \mid v \in V\}$, i.e., a literal is either a variable v or the negation of a variable $\overline{v}$. We define a clause as a subset of literals. Given an assignment $\psi\colon V \to \{true, false\}$, we can extend ψ on the set of literals by setting $\psi(\overline{v}) = true$ if $\psi(v) = false$ and $\psi(\overline{v}) = false$ if $\psi(v) = true$. We say that a clause c is satisfied under ψ if at least one of the literals in c is assigned the value true under ψ.*

An instance of the SAT *problem consists of a set of variables V and a set of clauses K. We call an assignment $\psi\colon V \to \{true, false\}$ feasible if all clauses in K are satisfied by ψ. The task is to decide whether there exists a feasible assignment.*

As an example an instance could consist of the set of variables $V = \{v_1, v_2, v_3, v_4\}$ and the set of clauses $K = \{\{\overline{v}_1, v_2, v_4\}, \{v_1, \overline{v}_2, v_3\}, \{v_1, \overline{v}_3, \overline{v}_4\}\}$. The instance is satisfiable since the assignment $\psi(v_1) = \psi(v_2) = \psi(v_3) = \psi(v_4) = true$ satisfies all clauses in K. We can also write the set of clauses K alternatively in the conjunctive normal form. In

the conjunctive normal form the set of clauses is replaced by the conjunction of its clauses and every clause is written as a disjunction of its literals. In our example this yields the Boolean formula $(\overline{v_1} \vee v_2 \vee v_4) \wedge (v_1 \vee \overline{v_2} \vee v_3) \wedge (v_1 \vee \overline{v_3} \vee \overline{v_4})$. Other variants of the SAT problem such as the 3SAT problem and the planar 3SAT problem have been proven to be NP-complete as well [35, 62]. The 3SAT problem is a variant of the SAT problem where every clause consists of at most 3 literals. The planar 3SAT problem which will be important in our NP-hardness proof of the minimum-error triangulation problem is a special case of the 3SAT problem in which the incidence graph of the instance is assumed to be planar.

Definition 4.1.2. *Given an instance of the* SAT *problem with variables V and clauses K, the vertex set of its incidence graph is given by the disjoint union of V and K and edges $\{v, c\}$ connecting a variable v to a clause c if c contains v or $\overline{v}$.*

Definition 4.1.3. *A Jordan arc is an injective continuous map $\sigma\colon [0,1] \to \mathbb{R}^2$. We call $\sigma(0)$ and $\sigma(1)$ the endpoints of σ. A planar embedding of the graph $G = (V, E)$ consists of*

1. *a map $f\colon V \to \mathbb{R}^2$ and*
2. *for every $e = \{s, t\} \in E$ a Jordan arc σ_e with endpoints $f(s)$ and $f(t)$, called the embedding of e,*

such that for all $e = \{s, t\} \in E$ and $e' \in E \backslash \{e\}$ we have $(\sigma_e([0,1]) \cap \sigma_{e'}([0,1])) \subset f(V)$ and $(\sigma_e([0,1]) \backslash \{f(s), f(t)\}) \cap f(V) = \emptyset$. Thus the embedded edge does not cross any point from $f(V)$ or any other embedded edge except at the embedding of its endpoints. We call a graph G planar if there exists a planar embedding of G.

Definition 4.1.4. *An instance of the planar* 3SAT *problem consists of a set of variables V and a set of clauses K such that K contains only clauses of cardinality 3 and the incidence graph of the instance is planar.*

Lichtenstein [62] shows that the planar 3SAT problem is NP-hard. Furthermore he shows that we can restrict ourselves to a planar embedding of the incidence graph of the following form: An instance of the planar 3SAT problem can be embedded into the plane such that every clause is represented by a vertex and every variable by a box placed on the horizontal axis. A box is connected to a vertex via a rectilinear edge if the respective variable is contained in the clause. For an example, see Figure 4.1. Such an embedding is also used, for example, in [57].

Before we present the NP-hardness proof we repeat the definition of the minimum-error triangulation problem.

4.2 Preliminaries

We recall the definition of the convex hull.

Definition 4.2.1. *For a set $M \subset \mathbb{R}^d$ let $\mathsf{conv}(M) = \{\sum_{i=1}^n \alpha_i x_i \mid \sum_{i=1}^n \alpha_i = 1, \alpha_i \geq 0, n \in \mathbb{N}, \{x_1, \ldots, x_n\} \subset M\}$ denote the convex hull of M. If $M = \{u, v\}$ we also denote the convex hull of u and v by $\overline{uv}$.*

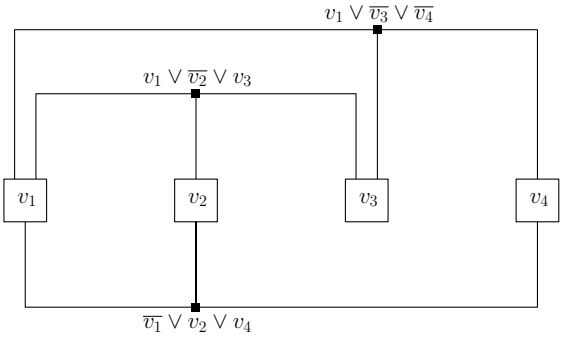

Figure 4.1: Embedding of the 3SAT formula $(\overline{v_1} \vee v_2 \vee v_4) \wedge (v_1 \vee \overline{v_2} \vee v_3) \wedge (v_1 \vee \overline{v_3} \vee \overline{v_4})$.

Remember that a triangulation of a point set $\mathcal{S} \subset \mathbb{R}^2$ is a maximal set of non-crossing straight line edges with endpoints in $\mathcal{S}$.

Definition 4.2.2. *A triangulation D of $\mathcal{S} \subset \mathbb{R}^2$ is a subset of $\{\overline{uv} \mid u, v \in \mathcal{S}, u \neq v\}$ such that for all $e, e' \in D$ we have $e \cap e' \subset \mathcal{S}$ and $e \cap \mathcal{S} = \{u, v\}$ for $e = \overline{uv}$. Furthermore D is maximal, i.e. there is no other edge from $\{\overline{uv} \mid u, v \in \mathcal{S}, u \neq v\}$ which can be added to D without violating this property. For three non-collinear points $s, t, u \in \mathbb{R}^2$ we call $T = \mathsf{conv}(s, t, u)$ a triangle with vertices s, t, u and edges $\overline{st}, \overline{us}, \overline{tu}$. We say that a triangle T is contained in a triangulation D of $\mathcal{S}$ and write $T \in D$ if its edges belong to the triangulation D and T does not contain points from $\mathcal{S}$ other then its vertices, i.e., $T \cap \mathcal{S} = \{s, t, u\}$.*

A minimum-error triangulation instance consists of the following parts. Let $\mathcal{S} \subset \mathbb{R}^2$ be a set of n points and $f \colon \mathcal{S} \to \mathbb{R}$. We call $\mathcal{S}$ the set of triangulation points and $f(s)$ the measurement value of $s \in \mathcal{S}$. Additionally, we are given a set $\mathcal{R} \subset \mathsf{conv}(\mathcal{S})$ of m points and a function $h \colon \mathcal{R} \to \mathbb{R}$. We refer to $\mathcal{R}$ as the set of reference points and to $h(r)$ as the reference value of $r \in \mathcal{R}$.

For a triangulation D of $\mathcal{S}$ we now define the linear interpolation of f with respect to D as follows. For every triangle $T \in D$ with vertices s, t, u and every point $v \in T$ written as a convex combination $v = \alpha s + \beta t + \gamma u$ we set $f_D(v) = \alpha f(s) + \beta f(t) + \gamma f(u)$. The minimum-error triangulation problem asks for a triangulation D of $\mathcal{S}$ that minimizes the squared error between the reference values and the interpolation f_D, i.e.,

$$\mathrm{Err}_D(\mathcal{R}) = \sum_{r \in \mathcal{R}} (f_D(r) - h(r))^2.$$

Furthermore we define the *zero-error triangulation problem* as the problem to decide whether there exists a triangulation D of $\mathcal{S}$ with $\mathsf{Err}_D(\mathcal{R}) = 0$ or alternatively $f_D(r) = h(r)$ for all $r \in \mathcal{R}$.

4.3 Overview of the Main Idea

We show that the minimum-error triangulation problem is NP-hard to approximate. To show this we first prove the NP-hardness of the closely related zero-error triangulation problem.

Theorem 4.3.1. *The zero-error triangulation problem is NP-hard.*

Notice that the NP-hardness of this problem directly implies that the minimum-error triangulation problem likely cannot be approximated efficiently.

Corollary 4.3.2. *The minimum-error triangulation problem cannot be approximated within any multiplicative factor in polynomial time unless* $\mathsf{P} = \mathsf{NP}$.

For every instance of the planar 3SAT problem we construct an instance for the zero-error triangulation problem by replacing the boxes, vertices and edges of its rectilinear embedding in the plane by a set of triangulation points and reference points. For this purpose we handle each component of the 3SAT embedding individually. We construct *variable gadgets*, which replace the boxes, *wire gadgets*, which replace the rectilinear edges, *clause gadgets*, which replace the vertices, and *negation gadgets*, which are inserted on wires between variables and clauses that use those variables in negated form[1]. The combination of these gadgets then constitutes an instance of the zero-error triangulation problem.

Let V be the set of variables and K the set of clauses of the planar 3SAT instance. We have to guarantee that every zero-error triangulation D of the triangulation instance corresponds to a feasible assignment of variables and vice versa. Since a variable can take only two values, *true* and *false*, we have to reflect this in the possible zero-error triangulations. We ensure that there are only two possible zero-error triangulations on the points that belong to a variable gadget and the negation and wire gadgets that are attached to it.

Definition 4.3.3. *Let* $C = \{y \mid \|x - y\|_2 = \rho\}$ *be a circle around a point* $x \in \mathbb{R}^2$ *with radius* $\rho \in \mathbb{R}_{\geq 0}$. *We denote with* $I_C = \{y \in \mathbb{R}^2 \mid \|x - y\|_2 < \rho\}$ *the* interior *of* C, *with* $O_C = \mathbb{R}^2 \backslash (C \cup I_C)$ *the* exterior *of* C, *and with* $B_C = I_C \cup C$ *the closed disk bounded by* C.

We will now describe the main tool in the construction of our gadgets. Suppose our triangulation instance consists of several reference points $\mathcal{R}$ and a set of triangulation points $\mathcal{S}$, which satisfy the following properties. For every $r \in \mathcal{R}$ there exists a circle C_r with $r \in I_{C_r}$ such that the respective disks are pairwise disjoint, i.e., $B_{C_r} \cap B_{C_{r'}} = \emptyset$ for $r, r' \in \mathcal{R}$ with $r \neq r'$. Furthermore for every $r \in \mathcal{R}$ there exist four distinct points $a_r, b_r, c_r, d_r \in C_r \cap \mathcal{S}$ such that $\overline{a_r c_r} \cap \overline{b_r d_r} = \{r\}$. All remaining points from $\mathcal{S}$ are contained in $\bigcap_{r \in \mathcal{R}} O_{C_r}$ and therefore lie outside all disks. We say that a triangle T represents a point $r \in \mathcal{R}$ with *zero error* if $r \in T$ and $h(r) = f_T(r)$. Our goal is now to set the reference values $h \colon \mathcal{R} \to \mathbb{R}$ and measurement values $f \colon \mathcal{S} \to \mathbb{R}$ such that every triangle that represents r with zero error has either $\overline{a_r c_r}$ or $\overline{b_r d_r}$ as an edge. For an illustration see Figure 4.2.

We achieve this goal by exploiting well-known geometric properties of the unit paraboloid in $\mathbb{R}^3$. For a set $M \subset \mathbb{R}^2$ we denote by $M' = \{(x_1, x_2, x_1^2 + x_2^2) \mid (x_1, x_2) \in M\}$ the lift of M onto the paraboloid. For every triangulation point $p = (p_1, p_2) \in \mathcal{S}$ we set $f(p) = p_1^2 + p_2^2$. For every reference point $r \in \mathcal{R}$ we consider the lifted circle C_r'. It is a known fact that there exists a plane E_r in $\mathbb{R}^3$ whose intersection with the paraboloid is exactly C_r'. Since the paraboloid is convex, we know furthermore that the lift of the interior I_{C_r}' onto the paraboloid lies below the plane E_r and the lift of the exterior O_{C_r}' onto the paraboloid lies above the plane. Thus for each reference point $r = (r_1, r_2)$ we do the following. We define $h(r)$ such that $(r_1, r_2, h(r)) \in E_r$. Since $\mathcal{S} \subset C_r \cup O_{C_r}$, no

[1] In fact, our clause gadget negates the first two variables in the clause by default, so a negation gadget must be used to undo the negation of one of the first two variables or to negate the third variable.

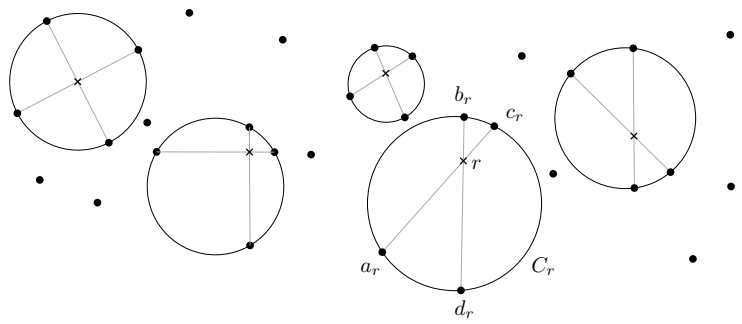

Figure 4.2: The reference points are depicted as crosses. The other points are triangulation points, lying either on or outside the circles.

point of the lift $\mathcal{S}'$ lies under E_r. Thus every triangle T all of whose vertices lie in O_{C_r} cannot triangulate r with zero error, since the lifted vertices would lie entirely above the plane E_r. In conclusion, a triangle T represents r with zero error if and only if it contains either the edge $\overline{a_r c_r}$ or the edge $\overline{b_r d_r}$. We will use these two options to encode the values *true* and *false* for the variables of the planar 3SAT instance.

However, in our construction of the triangulation instance we cannot require $B_{C_r} \cap B_{C_{r'}} = \emptyset$ for all $r, r' \in \mathcal{R}$ with $r \neq r'$. It is even crucial that some of these intersections are non-empty: by placing some of the points $a_{r'}, b_{r'}, c_{r'}$ and $d_{r'}$ on B_{C_r}, we achieve that the choice of a triangulation edge that represents r with zero error influences what triangulation edge can be chosen to represent r' with zero error. Such dependencies are what makes our gadgets work. We only have to be careful that the points that are additionally placed in B_{C_r} are placed such that they can never be used as vertices of triangles that represent r with zero error. In the next section we describe our approach more formally.

4.4 Notation and Local Properties

Our triangulation instance consists of a set of triangulation points with integral coordinates $\mathcal{S} \subset \mathbb{Z}^2$ and a set $\mathcal{R} \subset \mathsf{conv}(\mathcal{S})$ of reference points.

Measurement values: The measurement values are determined by the function $f\colon \mathbb{R}^2 \to \mathbb{R}$ with $f(p_1, p_2) = p_1^2 + p_2^2$.

Reference values: For every circle $C = \{y \mid \|x - y\|_2 = \rho\}$ around center $x = (x_1, x_2) \in \mathbb{R}^2$ with radius ρ, we define the function $h_C\colon \mathbb{R}^2 \to \mathbb{R}$ with $h_C(r_1, r_2) = 2x_1 r_1 + 2x_2 r_2 - x_1^2 - x_2^2 + \rho^2$. For every $r \in \mathcal{R}$ we choose one circle C and set $h(r) = h_C(r)$.

Thus the measurement values are fixed while we leave some freedom to choose the reference values during the construction of the instance. Observe that the function graph of f is the unit paraboloid $\{(p_1, p_2, p_1^2 + p_2^2) \mid (p_1, p_2) \in \mathbb{R}^2\}$ and the function graph of h_C is the plane containing the lift of C onto the paraboloid (see Figure 4.3).

Coupled circle: Every point $r \in \mathcal{R}$ is *coupled* to a circle, which we denote by C_r. It will be defined during the construction of the gadgets and determines the reference value $h(r) = h_{C_r}(r)$.

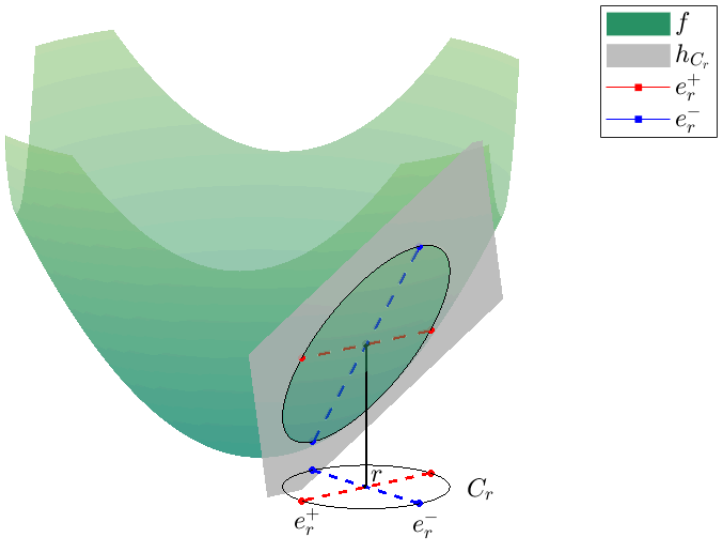

Figure 4.3: Example of a reference point r with coupled circle C_r and its positive/negative edges crossing at r. Lifting the red and blue points to $\mathbb{R}^3$, with their measurement values as third coordinate, we see that these points lie on both the paraboloid and the plane which contains $(r, h_{C_r}(r))$ and the lift of C_r.

Positive and negative edge: Every point $r \in \mathcal{R}$ comes with a positive and a negative edge called e_r^+ or e_r^-, respectively[2]. They intersect each other at r (i.e., $e_r^+ \cap e_r^- = \{r\}$) and their vertices are contained in the coupled circle C_r. It will always hold that e_r^+ has positive slope and e_r^- has negative slope.

On the final triangulation instance we require that every zero-error triangulation D corresponds to a feasible assignment on the variables V of the planar 3SAT instance. For this purpose we define the positive and negative signals which later correspond to setting the value of a variable to *true* or $false$.

Positive and negative signal: We say for a triangulation D that the *signal* at $r \in \mathcal{R}$ is *positive* if D contains edge e_r^+ and *negative* if it contains e_r^-, otherwise we call it *ambiguous*. Similarly for every set $M \subset \mathcal{R}$ we call D *positive* on M if the signal is positive at all $r \in M$ and *negative* on M if the signal is negative at all $r \in M$.

Definition 4.4.1. *Let D be a triangulation of $\mathcal{S}$ and $M \subset \mathcal{R}$. We define the error incurred by D on M as*

$$\mathsf{Err}_D(M) = \sum_{r \in M} (f_D(r) - h(r))^2.$$

Definition 4.4.2. *We say that $r \in \mathcal{R}$ is* represented with ϵ error *by a triangle T, if $r \in T$ and $(f_T(r) - h(r))^2 = \epsilon$. Here f_T denotes the linear interpolation of f on T. In particular we say that r is represented with zero error by T if $f_T(r) = h(r)$.*

[2]More precisely this is true for all reference points except some reference points in the clause and negation gadget. We elaborate on this during the construction of the gadgets.

Lemma 4.4.3. *Let r be a point of $\mathcal{R}$ and let $T \subset \mathbb{R}^2$ be a triangle with vertices $s, t, u \in \mathcal{S}$ and $r \in \mathsf{conv}(\{s, t, u\} \cap C_r)$. Then r is represented with zero error by T.*

If the planar 3SAT instance is satisfiable, we argue that there is a triangulation containing one of e_r^+, e_r^- for every reference point r of the triangulation instance. Lemma 4.4.3 states that such a triangulation has in fact zero error (see also Figure 4.3). To represent r with zero error in any other way, we need at least one triangulation point inside and one outside C_r. This follows from the convexity of f.

Lemma 4.4.4. *Let $T \subset \mathbb{R}^2$ be a triangle with vertices $s, t, u \in \mathcal{S}$ representing $r \in \mathcal{R}$ with zero error. If $r \notin \mathsf{conv}(\{s, t, u\} \cap C_r)$, then $\{s, t, u\}$ has a non-empty intersection with I_{C_r} and O_{C_r}.*

We prove Lemma 4.4.3 and Lemma 4.4.4 in Section 4.8. We guarantee during the construction that only few triangulation points lie in I_{C_r} for each reference point r. With a concise case analysis we rule out that any of them can be used together with a point in O_{C_r} to form a triangle that represents r with zero error, which limits the choice to triangles containing one of e_r^+, e_r^-. This ensures that every zero-error triangulation yields a solution to the planar 3SAT instance.

4.5 The Gadgets

Before we give a formal definition of the gadgets we give a brief overview over the main constructions as well as their functionality. We start with describing the smallest construction called *bit*, the next larger constructions called *segments* up to the *gadgets*.

Bit: A small construction at a point $r \in \mathbb{Z}^2$. It contains r as a reference point and several triangulation points. Every triangulation of $\mathcal{S}$ will have either positive or negative signal at r. There are vertical and horizontal bits. Bits with the same orientation can be combined into more complex constructions, whereas vertical and horizontal bits cannot be combined directly.

Wire segment: A construction connecting two points $x, y \in \mathbb{Z}^2$ lying on the same vertical or horizontal line. It is built from vertical/horizontal bits. It transports a positive/negative signal from the bit at x to the bit at y or vice versa.

Multiplier segment: A construction centered at a point $x \in \mathbb{Z}^2$. It consists of two vertical and two horizontal bits and some additional points. It has two functions. First it serves as a multiplier of a signal: if one of the bits carries a positive/negative signal the other three bits will carry the same signal. Secondly it serves as a connection between vertical and horizontal bits.

Wire gadget: An extension of the wire segment to deal with arbitrary points $x, y \in \mathbb{Z}^2$. It consists of one horizontal wire segment, one vertical wire segment and a multiplier segment connecting both. It has the same functionality as the wire segment, namely transporting a positive/negative signal from the bit at x to the bit at y or vice versa.

Variable gadget: It serves as a representation of a variable. If ℓ is the number of clauses in the planar 3SAT instance, then this gadget consists of ℓ multiplier segments which are connected to each other via wire segments. It will carry a consistent

positive/negative signal. Furthermore it has 2ℓ outputs where we can attach wire gadgets to transport the signal of this variable gadget to clause gadgets.

Clause gadget: It serves as a representation of a clause. It has three inputs where the signals of three variable gadgets arrive though wire gadgets. It can be triangulated with zero error if and only if one of the first two signals is negative or the third signal is positive.

Negation gadget: Since a clause may contain negated variables, we need to transform a positive signal into a negative and vice versa. This is the functionality of the negation gadget: it has one input carrying a positive or a negative signal and one output carrying the opposite signal.

Every construction Z consists of a 5-tuple $(\mathcal{R}(Z), \mathcal{S}(Z), \mathcal{F}(Z), \mathcal{A}(Z), E(Z))$. Here $\mathcal{R}(Z)$ denotes the set of reference points, $\mathcal{S}(Z)$ denotes the set of triangulation points and the other three parts are the following.

Forbidden points: A set of points $\mathcal{F}(Z)$ that are not allowed to be triangulation points in the triangulation instance as a whole, i.e., $\mathcal{F}(Z) \cap \mathcal{S} = \emptyset$.

Anchor points: A set of anchor points $\mathcal{A}(Z) \subset \mathcal{R}(Z)$ which indicates where the construction can be combined with other constructions.

Mandatory edges: A set of edges $E(Z)$ with vertices in $\mathcal{S}(Z)$ which are assumed to be contained in any zero-error triangulation of $\mathcal{S}$.

Note that the zero-error triangulation problem as defined in Section 4.2 does not allow us to specify mandatory edges. We will deal with this in Section 4.6, where we explain how we can add reference points to our zero-error triangulation instances in such a way that any zero-error triangulation must contain all mandatory edges.

Construction of the gadgets

In the construction of the gadgets we make use of the following simplified notation. For points $x = (x_1, x_2), y = (y_1, y_2) \in \mathbb{R}^2$ we denote by $x \pm y$ the two points $x + y, x - y$ and by $x + (\pm y_1, y_2)$ the two points $x + (y_1, y_2), x + (-y_1, y_2)$. We define $x + (y_1, \pm y_2)$ and $x + (\pm y_1, \pm y_2)$ analogously.

We start by describing the construction of a bit at point $r \in \mathbb{Z}^2$ (see Figure 4.4). A bit can be oriented either horizontally or vertically. In the first case we denote the construction by b_r^h and in the second case by b_r^v. We first describe the construction of the horizontal bit.

We set $\mathcal{R}(b_r^h) = \{r\}$, thus r is the only reference point of this construction. The point r is coupled to a circle C_r which is centered on r and has radius $\sqrt{2}$. The integer grid points on this circle, that is, the points $r + (\pm 1, \pm 1)$, are triangulation points. Moreover, $r + (0, 1)$ and $r + (0, -1)$ are triangulation points, whereas $r + (-2, 0), r + (-1, 0), r +$

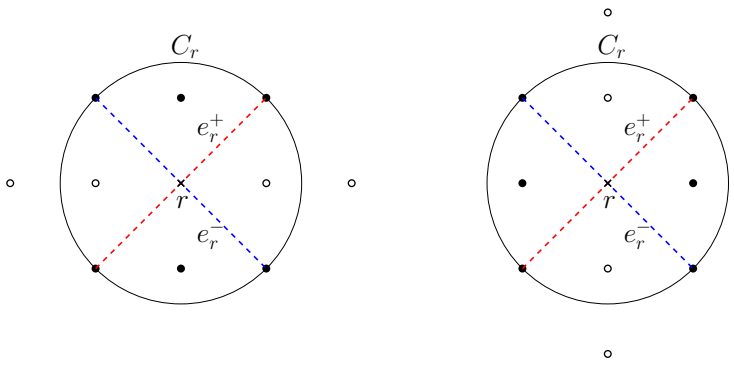

Figure 4.4: The (horizontal/vertical) bit at r with the positive edge in red and the negative edge in blue. The black points are triangulation points and the white points are forbidden.

$(1,0)$ and $r+(2,0)$ are forbidden points. In conclusion

$$\begin{aligned}
\mathcal{R}(b_r^h) &= \{r\} \\
\mathcal{S}(b_r^h) &= \{r+(\pm 1,\pm 1), r+(0,1), r+(0,-1)\} \\
\mathcal{F}(b_r^h) &= \{r+(-2,0), r+(-1,0), r+(1,0), r+(2,0)\} \\
\mathcal{A}(b_r^h) &= \emptyset \\
E(b_r^h) &= \emptyset.
\end{aligned}$$

Furthermore we define the positive and negative edge as

$$e_r^+ = \mathsf{conv}(r+(-1,-1), r+(1,1)), \quad e_r^- = \mathsf{conv}(r+(-1,1), r+(1,-1)).$$

As $r+(\pm 1,\pm 1) \in C_r$, any triangle containing either e_r^+ or e_r^- represents r with zero error by Lemma 4.4.3. For the vertical bit we reflect the whole construction at a line of slope 1 through r. Thus the definition of reference points, anchor points, mandatory edges as well as the definition of e_r^+, e_r^- and the coupled circle do not change; note that also in the vertical bit, e_r^+ has positive slope whereas e_r^- has negative slope. The set of triangulation points and forbidden points is now given by

$$\begin{aligned}
\mathcal{S}(b_r^v) &= \{r+(\pm 1,\pm 1), r+(1,0), r+(-1,0)\} \\
\mathcal{F}(b_r^v) &= \{r+(0,-2), r+(0,-1), r+(0,1), r+(0,2)\}.
\end{aligned}$$

Figure 4.4 illustrates both constructions. We show that the bit is well-behaved in the sense that any zero-error triangulation must carry either a positive or a negative signal at r. We show this by analyzing all possible triangles formed by points on the integer grid that represent r with zero error. Before we analyze this, we first observe that the error at a reference point does not change under an orthogonal transformation (i.e. rotation around the origin or reflection at a line) or translation $g\colon \mathbb{R}^2 \to \mathbb{R}^2$. More explicitly let r be a reference point, C_r its coupled circle and T a triangle with $r \in T$. The error of $g(T)$ at $g(r)$ with respect to the new measurement and reference values obtained after application of g on r, C_r and T equals the error of T at r.

Lemma 4.5.1. *Let $T \subset \mathbb{R}^2$ be a triangle representing $r \in \mathcal{R}$ with error ϵ. Let $g\colon \mathbb{R}^2 \to \mathbb{R}^2$ be an orthogonal transformation or translation. Let $\overline{T} = g(T), \overline{r} = g(r)$ and $\overline{C}_r =$*

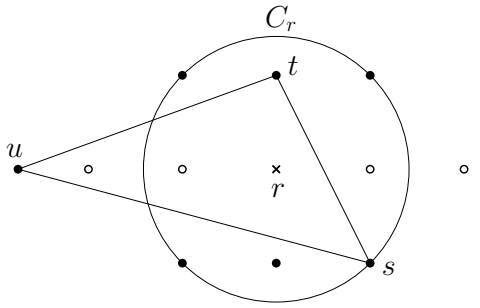

Figure 4.5: Here we see an example of a triangle T which contains r. Since $v(T) \cap C_r$ contains only s we clearly have $r \notin \mathsf{conv}(v(T) \cap C_r)$.

$g(C_r)$. *Then* $\overline{T}$ *is a triangle containing* $\overline{r}$, $\overline{C}_r$ *is a circle containing* $\overline{r}$ *in its interior and* $(f_{\overline{T}}(\overline{r}) - h_{\overline{C}_r}(\overline{r}))^2 = \epsilon$.

We defer the proof to Section 4.8.

Lemma 4.5.2. *Suppose the instance contains a bit at* r. *If* $\mathcal{S} \subset \mathbb{Z}^2$ *and* $\mathcal{S}$ *does not contain forbidden points of the bit, any triangulation* D *of* $\mathcal{S}$ *with* $\mathsf{Err}_D(r) = 0$ *contains one of* e_r^+, e_r^-.

Proof. Clearly, D can contain at most one of e_r^+ and e_r^-. We need to prove that we cannot have a triangle T that represents r with zero error and contains neither e_r^- nor e_r^+ as one of its edges. We first observe that we can assume $r = (0, 0)$ as, by Lemma 4.5.1, translation of T, r and C_r by $-r$ does not change the error.

Let $v(T) = \{s, t, u\} \subset \mathcal{S}$ denote the vertices of T. Notice that C_r does not contain any integral point aside from the vertices of e_r^+, e_r^-. Thus if $r \in \mathsf{conv}(v(T) \cap C_r)$, then e_r^+ or e_r^- would be an edge of T. So, henceforth, we consider the case that $r \notin \mathsf{conv}(v(T) \cap C_r)$. For an illustration we refer to Figure 4.5. Lemma 4.4.4 now tells us that $v(T) \cap I_{C_r} \neq \emptyset$. Since the construction and the error are invariant under rotation (except the labeling of e_r^+ and e_r^-, which may be switched) by Lemma 4.5.1, we may assume, without loss of generality, that the point $t = (0, 1)$ is included in $v(T) \cap I_{C_r}$.

We lift our construction into the 3-dimensional space. That is, for a point $p = (p_1, p_2) \in \mathbb{R}^2$, we denote with $p' = (p_1, p_2, f(p))$ its lift on the paraboloid $\Gamma_f = \{(q_1, q_2, q_1^2 + q_2^2) \mid (q_1, q_2) \in \mathbb{R}^2\}$ and analogously we define the lift of a set $M \subset \mathbb{R}^2$ as $M' = \{p' \mid p \in M\}$. Let E denote the plane that contains $v(T)'$. As $(0, 1) \in v(T)$ we know that $(0, 1, 1) \in v(T)' \subset E$. Furthermore $(0, 0, h_{C_0}(0)) = (0, 0, 2) \in E$ as T represents $(0, 0)$ with zero error. Thus a point (x_1, x_2, x_3) on E must satisfy $2ax_1 - x_2 - x_3 + 2 = 0$, for some fixed a.

We have $v(T)' \subset E \cap \Gamma_f$ and thus the remaining points of $v(T)$ must lie on the circle described by $x_1^2 + x_2^2 = 2ax_1 - x_2 + 2$. This is the circle C_T with center $(a, -1/2)$ and squared radius $a^2 + 9/4$. Since the construction and the error are invariant under reflection (except the labeling of e_r^+, e_r^-), we may now assume, without loss of generality, that a is non-negative. Let $\widetilde{T}$ denote the convex hull of $v(T)'$. As $\widetilde{T}$ must include r', T must include at least one point $u = (u_1, u_2)$, different from t, such that $u_1 \leq 0$. We will now investigate all possible locations of u. Let $g_a = \{(a, -\frac{1}{2}) \mid a \geq 0\}$ be the line segment containing all possible locations for the center of C_T. The gray area in

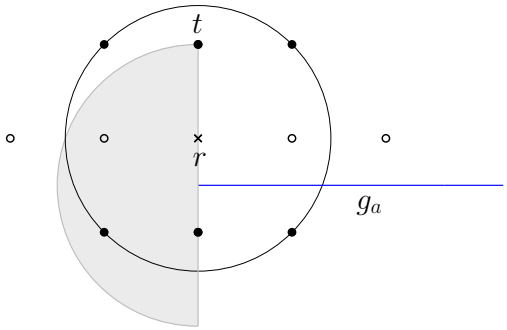

Figure 4.6: Here we see the line segment g_a in blue which contains all possible locations for the center of C_T. The gray area indicates all points which lie left from r on a circle through t with center in g_a. As we can see there are only few integral points lying in this area.

Figure 4.6 contains all possibilities for u, i.e. all points whose first coordinate is smaller or equal to zero and which are lying on a circle through t with center in g_a.

Remember that $\mathcal{S} \subset \mathbb{Z}^2$ so u must have integral coordinates.

Case 1: Suppose that $u_1 \leq -2$. The first coordinate of any point of C_T is at least $a - \sqrt{a^2 + 9/4}$. For $a \geq 0$, this expression grows with a, starting from $-3/2$ for $a = 0$. Thus such u cannot exist. We can also see in Figure 4.6 that the gray area does not contain such points.

Case 2: Suppose that $u_1 = -1$. We see in Figure 4.6 that the gray area only contains the points $(-1, 0), (-1, -1)$ as possibilities for u. More formally the circle equation reads $1 + u_2^2 = -2a - u_2 + 2$, so $(u_2 + 1/2)^2 = 5/4 - 2a$. For $a \geq 0$, this implies $|u_2 + 1/2| < \sqrt{5/4}$, and therefore the only candidate for u is $(-1, -1)$ (as $(-1, 0)$ is a forbidden point), with $a = 1/2$. Note that u lies on C_r. Now the first coordinate of third point s in $v(T)$ must be non-negative. Furthermore s has to lie on C_T, that is, on the circle with center $(1/2, -1/2)$ and radius $\frac{1}{2}\sqrt{10}$. Here the only candidates with integer coordinates are $(1, 1)$ (but then e_r^+ would be an edge of T), $(2, 0)$ (which is forbidden and thus not in $\mathcal{S}$), $(2, -1), (1, -2)$ and $(0, -2)$ (which are all invalid because T would then contain a fourth triangulation point $(0, -1)$). Therefore we cannot have $u_1 = -1$. In Figure 4.7 we see the circle C_T in blue as well as all possibilities for the third point s.

Case 3: Finally, suppose that $u_1 = 0$. Now we must have $u = (0, -1)$, since $u_2 = 0$ yields $u = r$, $u_2 > 0$ would imply that T does not contain r, whereas $u_2 < -1$ would imply that T contains $(0, -1)$ as a fourth triangulation point. But if $t' = (0, 1, 1)$ and $u' = (0, -1, 1)$ are both vertices of $\widetilde{T}$, then $r' = (0, 0, 2) \notin \widetilde{T}$ and thus we obtain a non-zero error at r. Therefore we cannot have $u_1 = 0$.

It follows that every triangle T that represents r with zero error contains either e_r^+ or e_r^-. □

The next larger components are the *wire segment* and the *multiplier segment*, which we build from bits. A *wire segment* w_{xy} connects two points $x, y \in \mathbb{Z}^2$ lying on the same horizontal or vertical line. The anchor points of this segment are x and y. Let

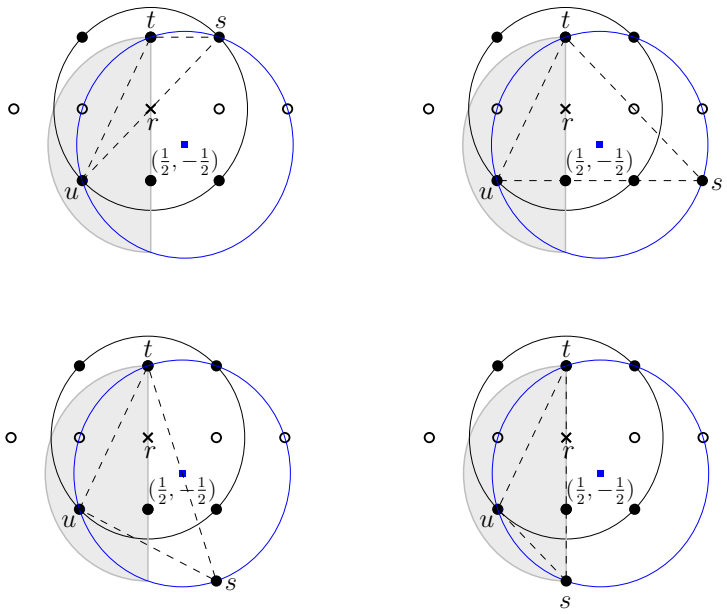

Figure 4.7: Here we see the circle C_T in blue which contains the points $t = (0, 1)$ and $u = (-1, -1)$. The last triangulation point s has to be an integral point on C_T which is not forbidden. Here we see the four possible choices for s and the resulting triangles.

$M = \{x + \lambda(y - x) \mid 0 \leq \lambda \leq 1\} \cap \mathbb{Z}^2$ be the set of all integral points lying on the line between x and y. If x and y lie on the same horizontal line we place a horizontal bit on all points in M. In conclusion

$$\mathcal{S}(w_{xy}) = \bigcup_{r \in M} \mathcal{S}(b_r^h) \qquad \mathcal{F}(w_{xy}) = \bigcup_{r \in M} \mathcal{F}(b_r^h) \qquad \mathcal{R}(w_{xy}) = M$$
$$\mathcal{A}(w_{xy}) = \{x, y\} \qquad E(w_{xy}) = \emptyset.$$

If x and y lie on the same vertical line we place a vertical bit on all points in M. In this case we replace b_r^h by b_r^v in the construction above to obtain the construction of w_{xy}. For an illustration of the wire segment we refer to Figure 4.8.

Given some set $M \subset \mathcal{R}$ remember that a triangulation D is called positive on M if D contains the edge e_r^+ for all $r \in M$ and negative on M if it contains the edge e_r^- for all $r \in M$.

Lemma 4.5.3. *Suppose the instance contains a wire segment and let $\widetilde{\mathcal{R}}$ be the reference points of this segment. If $\mathcal{S} \subset \mathbb{Z}^2$ and $\mathcal{S}$ does not contain forbidden points of the segment, any triangulation D of $\mathcal{S}$ with $\mathsf{Err}_D(\widetilde{\mathcal{R}}) = 0$ is either positive or negative on $\widetilde{\mathcal{R}}$.*

Proof. The wire segment connecting the points $(x_1, x_2), (y_1, y_2) \in \mathbb{Z}^2$ is completely built from bits. By Lemma 4.5.2 such a bit must have either a positive or a negative signal at its reference point. It is left to show that the signal is either positive or negative on the complete segment. Suppose this is not the case and that $x_1 = y_1$ (the other case follows analogously). Then there must be two reference points $r, q \in \widetilde{\mathcal{R}}$ with $r = q + (0, 1)$ where the signal at r differs from the signal at q. This is not possible as e_r^+ and e_q^- intersect each other and so do e_r^- and e_q^+. □

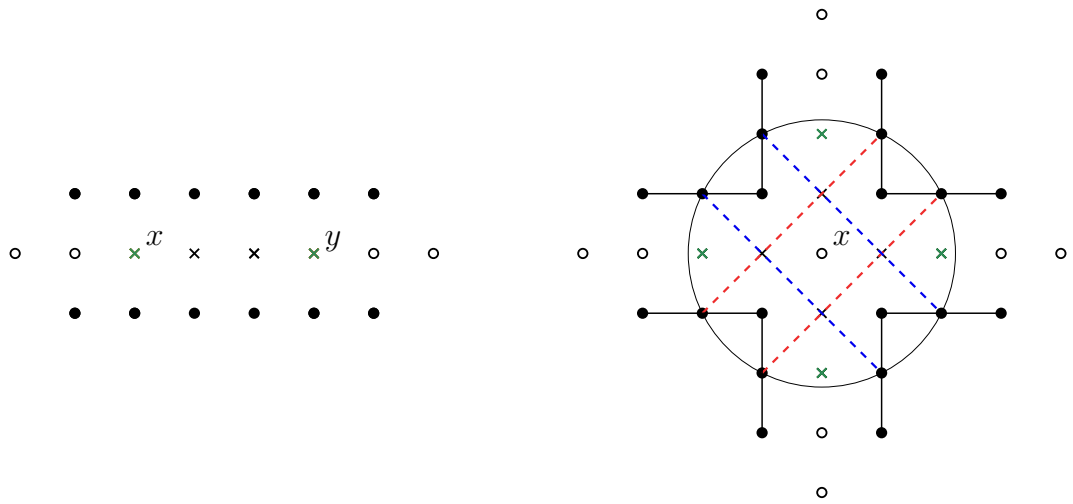

Figure 4.8: Example of a horizontal wire segment on the left and a multiplier segment with mandatory edges on the right. The red or blue edges indicate the positive or negative edges of the crossing points, respectively. All white points and all reference points are forbidden. The green points are anchor points.

The *multiplier segment* m_x at a point $x \in \mathbb{Z}^2$ consists of two horizontal bits at $x \pm (2,0)$ and two vertical bits at $x \pm (0,2)$. These four points are simultaneously anchor points. Furthermore we add four inner reference points $x \pm (0,1), x \pm (1,0)$ whose coupled circle is of radius $\sqrt{5}$ and centered around x. So the circle contains the points $x + (\pm 2, \pm 1), x + (\pm 1, \pm 2)$. The positive/negative edges of the inner reference points and the mandatory edges are shown in Figure 4.8. In conclusion

$$\begin{aligned}
\mathcal{R}(m_x) &= \{x \pm (2,0), x \pm (0,2), x \pm (0,1), x \pm (1,0)\} \\
\mathcal{S}(m_x) &= \mathcal{S}(b^h_{x+(2,0)}) \cup \mathcal{S}(b^h_{x-(2,0)}) \cup \mathcal{S}(b^v_{x+(0,2)}) \cup \mathcal{S}(b^v_{x-(0,2)}) \\
\mathcal{F}(m_x) &= \mathcal{F}(b^h_{x+(2,0)}) \cup \mathcal{F}(b^h_{x-(2,0)}) \cup \mathcal{F}(b^v_{x+(0,2)}) \cup \mathcal{F}(b^v_{x-(0,2)}) \\
\mathcal{A}(m_x) &= \{x \pm (2,0), x \pm (0,2)\} \\
E(m_x) &= \{\mathsf{conv}(x + (i,j), x + (i+1,j)) \mid i \in \{-3,-2,1,2\}, j \in \{-1,1\}\} \\
&\quad \cup \{\mathsf{conv}(x + (j,i), x + (j,i+1)) \mid i \in \{-3,-2,1,2\}, j \in \{-1,1\}\}.
\end{aligned}$$

Figure 4.8 shows the multiplier segment.

Lemma 4.5.4. *Suppose the instance contains a multiplier segment and let $\widetilde{\mathcal{R}}$ be the reference points of this segment. If $\mathcal{S} \subset \mathbb{Z}^2$ and $\mathcal{S}$ does not contain forbidden points of the segment, any triangulation D of $\mathcal{S}$ with $\mathsf{Err}_D(\widetilde{\mathcal{R}}) = 0$ is either positive or negative on $\widetilde{\mathcal{R}}$.*

Proof. We consider a multiplier segment at a point $x \in \mathbb{Z}^2$. By Lemma 4.5.1 we can assume that $x = (0,0)$. We use Lemma 4.5.2 to see that the signal on the reference points of bits must be either positive or negative. Thus D must contain one of the edges e_r^+, e_r^- for every reference point $r \in \{\pm(0,2), \pm(2,0)\}$. Let F be any set of edges that consists of the mandatory edges of the multiplier segment and at least one of the edges e_r^+, e_r^- for each $r \in \{\pm(0,2), \pm(2,0)\}$. These edges isolate the inner reference points

from the remaining instance, as every triangle that contains one of the inner reference points and a point outside of the segment must intersect at least one of the edges of F, regardless which of the sixteen possibilities for F is chosen.

Let T be a triangle in D representing an inner reference point r with zero error. The multiplier segment is invariant under rotation by $\frac{\pi}{2}$ (except the labeling of positive and negative edges). Furthermore rotation does not change the error at r by Lemma 4.5.1. Thus we can fix r to be $(-1, 0)$.

We claim that T contains one of e_r^+, e_r^- as an edge. We already observed that the vertices $v(T)$ of T consist of triangulation points from the multiplier segment at $(0, 0)$. If $r \in \mathsf{conv}(v(T) \cap C_r)$ we see that one of e_r^+, e_r^- is an edge of T. If this is not the case we apply Lemma 4.4.4 to see that $v(T) \cap I_{C_r} \neq \emptyset \neq v(T) \cap O_{C_r}$, that is, T must have at least one vertex strictly inside the circle and at least one vertex strictly outside the circle. We enumerate all possibilities for such T.

Case 1: Assume that $t = (-1, 1) \in v(T)$. Then $(-1, -1) \notin v(T)$, as $h_{C_r}(r) = 5 \neq 2 = \frac{1}{2}(f(-1, 1) + f(-1, -1))$. Figure 4.9 shows all possibilities to choose the second point u of $v(T)$ such that $\mathsf{conv}(r, t, u) \backslash \{r, t, u\}$ does not contain triangulation points or intersect mandatory edges. Among these points there are four points from O_{C_r}. As we know that $v(T)$ must contain at least one of these, we consider all the cases where we choose one of them as the second point u. Figure 4.9 shows all possibilities choosing the last point of $v(T)$ depending on the choice of u. We see in Figure 4.9 that in the sub-cases (1.1) and (1.2) we have $r \notin T$. In sub-case (1.3) we have $v(T) = \{(-1, 1), (-2, 1), (1, -3)\}$ and $r = \frac{1}{4}(-1, 1) + \frac{1}{2}(-2, 1) + \frac{1}{4}(1, -3)$ but

$$h_{C_r}(r) = 5 \neq \frac{11}{2} = \frac{2}{4} + \frac{5}{2} + \frac{10}{4} = \frac{1}{4}f(-1, 1) + \frac{1}{2}f(-2, 1) + \frac{1}{4}f(1, -3).$$

In sub-case (1.4) we have $v(T) = \{(-1, 1), (-3, -1), (1, 1)\}$ and $r = \frac{1}{2}(-3, -1) + (1, 1)$ but

$$h_{C_r}(r) = 5 \neq 6 = \frac{1}{2}(f(-3, -1) + f(1, 1)).$$

Thus in both cases we get a contradiction to T representing r with zero error.

Case 2: For $t = (1, 1) \in v(T)$ we do the same and obtain two possibilities to choose a point from O_{C_r}. Both are depicted in Figure 4.9. In sub-case (2.1) we have $r \notin T$. In sub-case (2.2) we calculate as in case 1 that T does not represent r with zero error.

The remaining cases $t = (\pm 1, -1)$ can be shown analogously. The computations do not change as f is invariant under reflection. We conclude that D contains one of e_r^+, e_r^- for all $r \in \widetilde{\mathcal{R}}$ and must be either positive or negative on the whole gadget. □

To make the construction of gadgets easier we talk about the combination of multiple constructions. We can combine two constructions Z_1, Z_2 into a construction Z, if the following conditions are met

1. $\mathcal{A}(Z_1) \cap \mathcal{A}(Z_2) \neq \emptyset$,
2. $\mathcal{S}(Z_i) \cap (\mathcal{R}(Z_j) \cup \mathcal{F}(Z_j)) = \emptyset$ for $i, j \in \{1, 2\}$ with $i \neq j$,

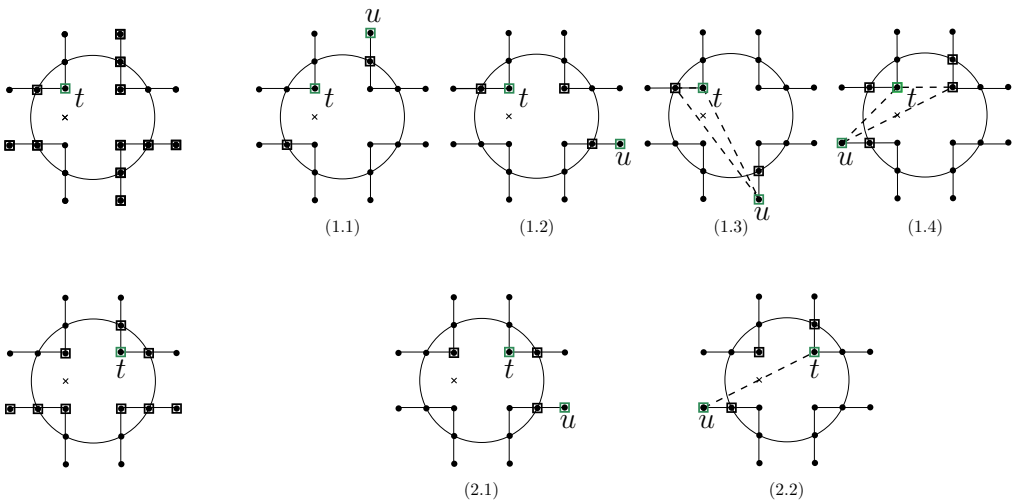

Figure 4.9: Possibilities to build $v(T)$ starting with $t = (\pm 1, 1)$. The points outlined in green are currently assumed to be in $v(T)$. All points that are not boxed cannot be in $v(T)$.

3. $(\bigcup_{e \in E(Z_i)} e) \cap (\mathcal{S}(Z_j) \backslash \mathcal{S}(Z_i)) = \emptyset$ for $i, j \in \{1, 2\}$ with $i \neq j$.

In words they must have at least one anchor point in common; the triangulation points of one construction are disjoint from the reference points and the forbidden points of the other; and the mandatory edges, excluding their vertices, of one construction are disjoint from the triangulation points of the other construction. We then define the combination Z of the two constructions Z_1 and Z_2 as the union of all of their components.

$$\mathcal{S}(Z) = \mathcal{S}(Z_1) \cup \mathcal{S}(Z_2) \qquad \mathcal{R}(Z) = \mathcal{R}(Z_1) \cup \mathcal{R}(Z_2) \qquad \mathcal{F}(Z) = \mathcal{F}(Z_1) \cup \mathcal{F}(Z_2)$$
$$\mathcal{A}(Z) = \mathcal{A}(Z_1) \cup \mathcal{A}(Z_2) \qquad E(Z) = E(Z_1) \cup E(Z_2).$$

To obtain the larger *variable gadget* and *wire gadget* we combine wire segments with multiplier segments. Remember that the wire gadget replaces the rectilinear edges of the 3SAT embedding. Let $x = (x_1, x_2), y = (y_1, y_2) \in \mathbb{Z}^2$ with $|x_1 - y_1| \geq 4$ and $|x_2 - y_2| \geq 4$. For the wire gadget w_x^y we place a multiplier segment on (x_1, y_2) to form a corner and for the wire gadget w_y^x we place a multiplier segment on (y_1, x_2) to form a corner. In both cases we combine the multiplier segment with two wire segments to connect it to x and y. For simplicity we assume that $x_1 < y_1$ and $x_2 < y_2$; the general case follows similarly. For w_x^y we combine the multiplier segment at (x_1, y_2) with the wire segment from x to $(x_1, y_2 - 2)$ and the wire segment from y to $(x_1 + 2, y_2)$. For w_y^x we combine the multiplier segment at (y_1, x_2) with the wire segment from x to $(y_1 - 2, x_2)$ and the wire segment from y to $(y_1, x_2 + 2)$.

A variable gadget v_x at $x \in \mathbb{Z}^2$ consists of ℓ multiplier segments at sufficiently large distance $\alpha \in \mathbb{Z}$, which we do not specify further. Here ℓ denotes the number of clauses of the planar 3SAT instance. Concretely, we place a multiplier segment on each of the points $x + (k\alpha, 0)$ with $0 \leq k \leq \ell - 1$ and combine them via wire segments from $x + (k\alpha + 2, 0)$ to $x + ((k + 1)\alpha - 2, 0)$ for $0 \leq k \leq \ell - 2$. The multiplier segments ensure that the gadget can later be connected at its anchor points to multiple clause gadgets. We observe that the described combinations of segments for both gadgets are feasible and that they have the following crucial property.

Lemma 4.5.5. *Suppose the instance contains a wire/variable gadget and let $\widetilde{\mathcal{R}}$ be the reference points of this gadget. If $\mathcal{S} \subset \mathbb{Z}^2$ and $\mathcal{S}$ does not contain forbidden points of the gadget, any triangulation D of $\mathcal{S}$ with $\mathsf{Err}_D(\widetilde{\mathcal{R}}) = 0$ is either positive or negative on $\widetilde{\mathcal{R}}$.*

Proof. The signal at a segment that is part of the gadget must be either positive or negative by Lemma 4.5.3 and Lemma 4.5.4. If it is connected to another segment at one of its anchor points, this anchor point determines the signal at both segments, which must agree with the signal at the anchor point. Proceeding like this we see that the signal must be either positive or negative on the whole gadget. □

Having the variable gadget and wire gadget in place we need two more constructions, namely the *clause gadget* and the *negation gadget*. Both are very similar to each other.

We explain how to build the clause gadget representing a clause of the form $\overline{v_1} \vee \overline{v_2} \vee v_3$ at a point $x \in \mathbb{Z}^2$. For simplicity we assume that $x = (0, 0)$. We declare $r = (0, 11)$ as a reference point and the points $\{(5, -15), (\pm 15, -5), (\pm 9, 13)\}$ as triangulation points. Notice that these triangulation points all lie on one circle centered at $(0, 0)$ with radius $\sqrt{250}$. This circle is the coupled circle C_r of r. The reference point r is special as it does not come with a positive and negative edge, instead we observe that it is represented with zero error by the following three triangles:

$$T_1 = \mathsf{conv}((5, -15), (9, 13), (-9, 13))$$
$$T_2 = \mathsf{conv}((15, -5), (9, 13), (-9, 13))$$
$$T_3 = \mathsf{conv}((-15, -5), (9, 13), (-9, 13))$$

This is true by Lemma 4.4.3 and $r \in T_i$ for $i = 1, 2, 3$.

A triangle T is blocked by an edge e if both cannot be part of the same triangulation of $\mathcal{S}$. This is the case if e is not an edge of T and $(e \cap T) \backslash \mathcal{S} \neq \emptyset$. Now we define for every $i \in \{1, 2, 3\}$ a reference point r_i with the crucial property that the triangle T_i is blocked by the positive edge of r_i for $i = 1, 2$ and by the negative edge of r_i for $i = 3$.

Let $a_1 = (-12, -17)$ and define r_1 to be the intersection of the two edges $e^+_{r_1} = \mathsf{conv}(a_1 + (1, -1), a_1 + (23, 4))$ and $e^-_{r_1} = \mathsf{conv}(a_1 + (1, 1), a_1 + (23, -4))$. Thus we have $r_1 = a_1 + (\frac{27}{5}, 0)$. We declare the vertices of $e^+_{r_1}$ and $e^-_{r_1}$ as triangulation points. Observe that they lie on a common circle C_{r_1}, which is the circle coupled to r_1. Furthermore we add three horizontal bits, one at each of the points $a_1 + (\ell, 0)$ for $\ell = 0, 1, 2$ and declare a_1 as anchor point. Let

$$\mathcal{R}_1 = \bigcup_{\ell=0}^{2} \mathcal{R}(b^h_{a_1+(l,0)}) \cup \{r_1\}$$
$$\mathcal{S}_1 = \bigcup_{\ell=0}^{2} \mathcal{S}(b^h_{a_1+(l,0)}) \cup \{a_1 + (1, \pm 1), a_1 + (23, \pm 4)\}$$
$$\mathcal{F}_1 = \bigcup_{\ell=0}^{2} \mathcal{F}(b^h_{a_1+(l,0)})$$
$$\mathcal{A}_1 = \{a_1\}$$
$$E_1 = \{\mathsf{conv}(a_1 + (1, -1), a_1 + (2, -1)), \mathsf{conv}(a_1 + (2, -1), a_1 + (3, -1)),$$
$$\mathsf{conv}(a_1 + (3, -1), a_1 + (23, -4)), \mathsf{conv}(a_1 + (23, -4), a_1 + (23, 4))\}$$

A similar construction is done at the anchor points $a_2 = (17, 12)$ and $a_3 = (-17, 12)$. Let $\gamma_2 \colon \mathbb{R}^2 \to \mathbb{R}^2$ be the reflection on the line with slope -1 through $(0, 0)$ and $\gamma_3 \colon \mathbb{R}^2 \to \mathbb{R}^2$

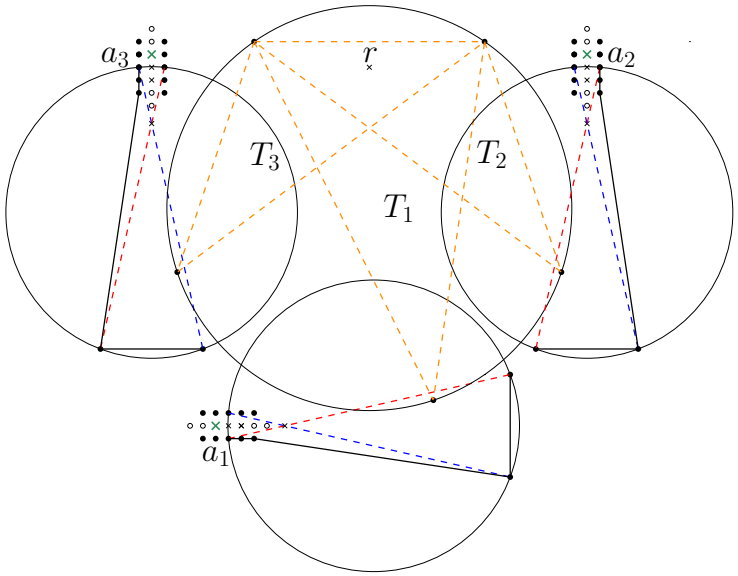

Figure 4.10: The clause gadget, where the red/blue edges indicate the positive/negative edges of the crossing points. The triangles T_1, T_2, T_3 are orange and the anchor points a_1, a_2, a_3 green.

the clockwise rotation by $\frac{3\pi}{2}$. Let $i = 2, 3$ and

$$\mathcal{R}_i = \gamma_i(\mathcal{R}_1) \quad \mathcal{S}_i = \gamma_i(\mathcal{S}_1) \quad \mathcal{F}_i = \gamma_i(\mathcal{F}_1) \quad \mathcal{A}_i = \{a_i\} \quad E_i = \gamma_i(E_1).$$

Notice that γ_2 does not change the slope while γ_3 changes the slope. Thus for every $r' \in \mathcal{R}_1$ we have $e^+_{\gamma_2(r')} = \gamma_2(e^+_{r'}), e^-_{\gamma_2(r')} = \gamma_2(e^-_{r'})$ and $e^+_{\gamma_3(r')} = \gamma_3(e^-_{r'}), e^-_{\gamma_3(r')} = \gamma_3(e^+_{r'})$.

In total we obtain

$$\mathcal{R}(c_x) = \bigcup_{i=1}^{3} \mathcal{R}_i \cup \{r\} \qquad \mathcal{S}(c_x) = \bigcup_{i=1}^{3} \mathcal{S}_i \cup \{(5, -15), (\pm 15, -5), (\pm 9, 13)\}$$

$$\mathcal{A}(c_x) = \bigcup_{i=1}^{3} \mathcal{A}_i \qquad E(c_x) = \bigcup_{i=1}^{3} E_i.$$

Furthermore we declare all points from $\bigcup_{i=1}^{3} \mathcal{F}_i$ and all non-triangulation points on or inside a circle coupled to a reference point $r' \in \mathcal{R}(c_x)$ as forbidden.

$$\mathcal{F}(c_x) = \bigcup_{i=1}^{3} \mathcal{F}_i \cup \bigcup_{r' \in \mathcal{R}(c_x)} B_{C_{r'}} \backslash \mathcal{S}(c_x)$$

The whole construction is depicted in Figure 4.10.

Lemma 4.5.6. *Suppose the instance contains a clause gadget and let $\widetilde{\mathcal{R}}$ be its reference points. If $\mathcal{S} \subset \mathbb{Z}^2$ and $\mathcal{S}$ does not contain forbidden points of the gadget, any triangulation D of $\mathcal{S}$ with $\mathsf{Err}_D(\widetilde{\mathcal{R}}) = 0$ must be negative on one of the anchor points a_1, a_2 or positive on a_3.*

Proof. The signal at a reference point of a bit must be either positive or negative by Lemma 4.5.2. This also includes the anchor points a_1, a_2, a_3.

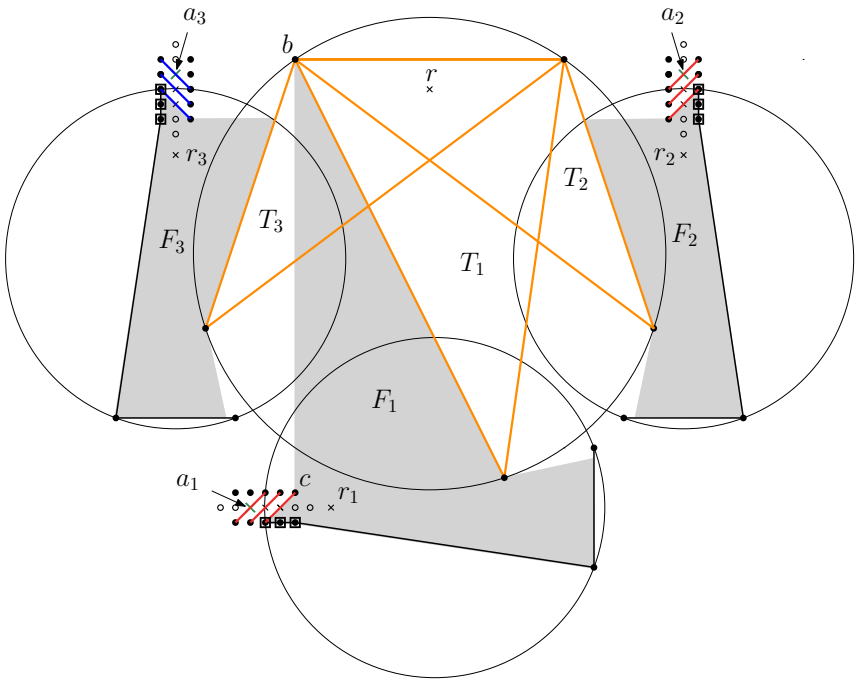

Figure 4.11: Here we see all three cases together. For $i \in \{1, 2, 3\}$, if we choose the first point t in $v(T)$ to be one of the three marked points near a_i, the remaining points of $v(T)$ must come from the shaded area F_i, otherwise T would intersect other triangulation points or edges. However, F_i does not contain any points outside C_{r_i} (note that $b = (-9, 13)$ lies just outside F_1, as $\mathsf{conv}(t, b, r_1)$ would include $c = (-9, -16)$ if t is any of the marked points near a_1).

Suppose that T_1 is in D and the signal at a_1 is positive. We show that the error at r_1 is positive contradicting the assumption that D is a zero-error triangulation.

Let T be the triangle in D with $r_1 \in T$ and let $v(T)$ denote its vertices. As T_1 belongs to D and the signal at $a_1 + (2, 0)$ is positive, we know that $e^+_{r_1}, e^-_{r_1}$ cannot be edges of T. Thus by Lemma 4.4.4 we know that $v(T)$ has a non-empty intersection with $I_{C_{r_1}}$ and $O_{C_{r_1}}$. Furthermore $v(T)$ must contain a point below the line that supports $e^-_{r_1}$, so $v(T)$ must contain at least one of the points $a_1 + (l, -1)$ for $l = 1, 2, 3$. Let t be this point. Choosing the second point u in $v(T)$ from $O_{C_{r_1}}$ already yields a contradiction, because for any choice of u from $O_{C_{r_1}}$, the hull $\mathsf{conv}(t, u, r_1)$ contains another triangulation point or intersects a mandatory edge, the triangle T_1, or the positive edge of $a_1 + (2, 0)$.

Analogously one can prove that T_2 and $e^+_{a_2}$ or T_3 and $e^-_{a_3}$ cannot be simultaneously in D. Figure 4.11 illustrates how to exclude all three combinations. Since r is triangulated with zero error by D one of the triangles T_1, T_2, T_3 must be in D. Thus the signal at one of a_1, a_2 must be negative or the signal at a_3 must be positive. □

The last gadget, the *negation gadget*, is constructed out of wires, multipliers and simplified clause gadgets. The core components of the negation gadget are the positive and negative negation segments. For the positive negation segment n^+_x at a point x we

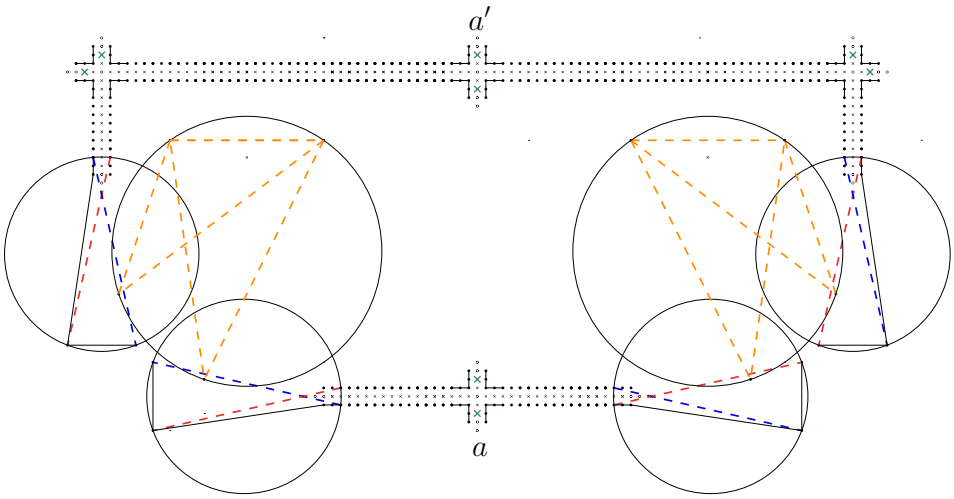

Figure 4.12: The negation gadget. If the signal at anchor point a is negative it is negated in the left segment. If the signal at a is positive it is negated in the right segment. Since the top wire carries a consistent signal, negation is ensured at a'.

follow the construction of the clause gadget, except that we omit the triangle T_3 and the construction at a_3. Thus following the notation for the clause gadget we define

$$\mathcal{R}(n_x^+) = \mathcal{R}(c_x)\backslash\mathcal{R}_3 \qquad \mathcal{S}(n_x^+) = \mathcal{S}(c_x)\backslash(\mathcal{S}_3 \cup \{x + (-15, -5)\}$$
$$\mathcal{A}(n_x^+) = \mathcal{A}(c_x)\backslash\mathcal{A}_3 \qquad E(n_x^+) = E(c_x)\backslash E_3$$

and define the forbidden points to be

$$\mathcal{F}(n_x^+) = \bigcup_{i=1}^{2} \mathcal{F}_i \cup \bigcup_{r' \in \mathcal{R}(n_x^+)} B_{C_{r'}}\backslash\mathcal{S}(n_x^+).$$

The negative negation segment n_x^- at a point x is now a reflection of the positive negation segment. Let $\gamma\colon \mathbb{R}^2 \to \mathbb{R}^2$ be the reflection at the vertical line through x. We define

$$\mathcal{R}(n_x^-) = \gamma(\mathcal{R}(n_x^+)) \qquad \mathcal{S}(n_x^-) = \gamma(\mathcal{S}(n_x^+)) \qquad \mathcal{F}(n_x^-) = \gamma(\mathcal{F}(n_x^+))$$
$$\mathcal{A}(n_x^-) = \gamma(\mathcal{A}(n_x^+)) \qquad E(n_x^-) = \gamma(E(n_x^+)).$$

The negation gadget n_x at a point $x \in \mathbb{Z}^2$ is a combination of several constructions. We place a multiplier segment at x, a positive negation gadget at $x + (27, 17)$ and a negative negation gadget at $x + (-27, 17)$. The anchor point $x + (2, 0)$ of the multiplier segment is then connected via a wire segment to the lower anchor point $x + (15, 0)$ of the positive negation segment. The anchor point $x - (2, 0)$ of the multiplier segment is connected via a wire segment to the lower anchor point $x - (15, 0)$ of the negative negation segment. Furthermore we place a multiplier segment at $x + (0, 38)$ and connect the anchor point $x + (2, 38)$ of this multiplier segment via a wire gadget to the upper anchor point $x + (44, 29)$ of the positive negation segment. Finally we connect the anchor point $x + (-2, 38)$ of this multiplier segment via a wire gadget to the upper anchor point $x + (-44, 29)$ of the negative negation segment. Figure 4.12 visualizes the construction. We analyze the signal at the anchor points $a = x - (0, 2)$ and $a' = x + (0, 40)$:

Lemma 4.5.7. *Suppose the instance contains a negation gadget at $x \in \mathbb{Z}^2$ and let $\widetilde{\mathcal{R}}$ be the reference points of this gadget. Let $\mathcal{S} \subset \mathbb{Z}^2$ and assume $\mathcal{S}$ does not contain forbidden points of the gadget. Any triangulation D of $\mathcal{S}$ with $\mathsf{Err}_D(\widetilde{\mathcal{R}}) = 0$ is positive at a iff it is negative at a'.*

Proof. We consider the positive negation segment at point $x + (27, 17)$, which equals the clause gadget without the construction at a_3 and T_3. We borrow the notation from the clause gadget. As in the proof of Lemma 4.5.6 one can show that neither T_1 and $e^+_{r_1}$ nor T_2 and $e^+_{r_2}$ can simultaneously be in D. As one of T_1, T_2 is in D this means that at most one of $a_1 = x + (15, 0), a_2 = x + (44, 29)$ has a positive signal. Analogously at most one of the signals at the anchor points $a'_1 = x - (15, 0), a'_2 = x + (-44, 29)$ of the negative negation segment at $x + (-27, 17)$ is negative.

Suppose that the signal at a is positive. Then by Lemma 4.5.3 the signal at a_1 must also be positive. By the observation above the signal at a_2 must then be negative and so must be the signal at a' by Lemma 4.5.5. If the signal at a is negative the signal must be positive at a'_2 and a' following the same arguments. □

4.6 Replacing Mandatory Edges

Before we dedicate ourselves to the proof of Theorem 4.3.1, we need to discuss how we can enforce that the mandatory edges are part of any zero-error triangulation, since the original definition of the zero-error triangulation problem does not allow us to specify mandatory edges. To this end, we slightly modify the previously constructed gadgets/segments as follows. Let Z be a construction. For every edge $e = \overline{st} \in E(Z)$ we add the reference point $r_e = \frac{1}{2}(s + t)$ to $\mathcal{R}(Z)$. It is left to define the circle C_{r_e} coupled to r_e. Notice that we would like to enforce the edge e to be in *every* zero-error triangulation of the gadget. Suppose that

1. $\{s, t\} \subset C_{r_e}$ and
2. $B_{C_{r_e}}$ does not contain further triangulation points.

Then any triangle with vertices s, t represents r_e with zero error by Lemma 4.4.3 and any triangulation which does not contain e has positive error at r_e by Lemma 4.4.4.

If we have $t \in \{s \pm (1, 0), s \pm (0, 1)\}$ then we define C_{r_e} as the circle centered at r_e with radius $\frac{\|s-t\|_2}{2}$. In this case $B_{C_{r_e}} \setminus \{s, t\}$ does not contain integral points and therefore it does not contain any triangulation points. Thus 1. and 2. are both satisfied for C_{r_e}.

The mandatory edges that do not satisfy the above property are all part of the clause gadget and the negation gadget. Figure 4.13 shows how to define C_{r_e} for the mandatory edges in the clause gadget. In Figure 4.13 we see that we can again define C_{r_e} as the circle centered at r_e with radius $\frac{\|s-t\|_2}{2}$ for three out of six long mandatory edges in the gadget. For the other three long edges in the clause gadget let Q_1 and Q_2 be the two squares which contain e as one of their edges. One of these squares contains triangulation points of the gadget other than s, t, while the other one does not. Let Q_1 be the square which does not contain any triangulation points other than s, t. We define C_{r_e} as the circumcircle of Q_1. In Figure 4.13 we see that 1. and 2. are then both satisfied for C_{r_e}. Observe that a similar construction can be done for the long mandatory edges in the negation gadget.

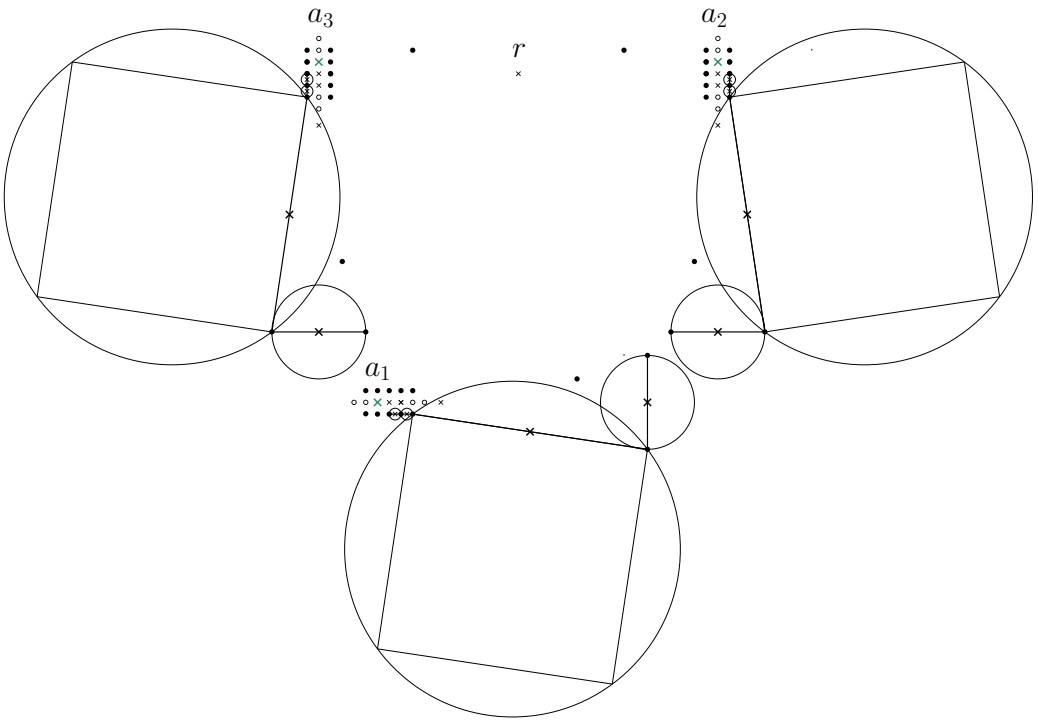

Figure 4.13: Here we see the replacement of mandatory edges in the clause gadget. For every mandatory edge $e = \overline{st}$ the set $B_{C_{r_e}}$ contains only s and t as triangulation points.

Finally we extend the set of forbidden points $\mathcal{F}(Z)$ by $B_{C_{r_e}} \setminus \{s, t\}$. The following corollary is an immediate consequence of Lemma 4.4.3 and Lemma 4.4.4.

Corollary 4.6.1. *Suppose the instance contains a modified construction with reference points $\widetilde{\mathcal{R}}$, including the reference points which replace the mandatory edges. If $\mathcal{S} \subset \mathbb{Z}^2$ and $\mathcal{S}$ does not contain forbidden points of the gadget, any triangulation D of $\mathcal{S}$ with $\mathsf{Err}_D(\widetilde{\mathcal{R}}) = 0$ contains all mandatory edges of this construction.*

4.7 The Reduction

Given an instance $\mathcal{I}$ of the planar 3SAT problem, with V the set of variables and K the set of clauses, we first explain how to construct the corresponding instance $\mathcal{I}_{\text{err}}$ of the zero-error triangulation problem. Let $k = |K| + |V|$. We fix an integral rectilinear embedding of the planar 3SAT instance on the plane, i.e., the vertices representing the clauses as well as the centers of the boxes representing the variables must have integral coordinates. We scale the embedding by an integer factor $\gamma \in O(k)$. Notice that the scaled embedding is still rectilinear and integral. Let $G(v)$ denote the center of the box belonging to a variable $v \in V$ and $G(c)$ the vertex belonging to a clause $c \in K$ of the scaled embedding. Recall that $G(v)$ lies on the horizontal axis for all $v \in V$.

The zero-error triangulation instance is constructed as follows: We place a variable gadget at $G(v)$ for all $v \in V$ and a clause gadget at $G(c)$ for all $c \in K$. For a clause $c \in K$ containing the variables v_1, v_2, v_3 we do the following: Notice that $G(v_1), G(v_2), G(v_3)$ lie on the horizontal axis and we assume that they appear on the axis from left to right in this order.

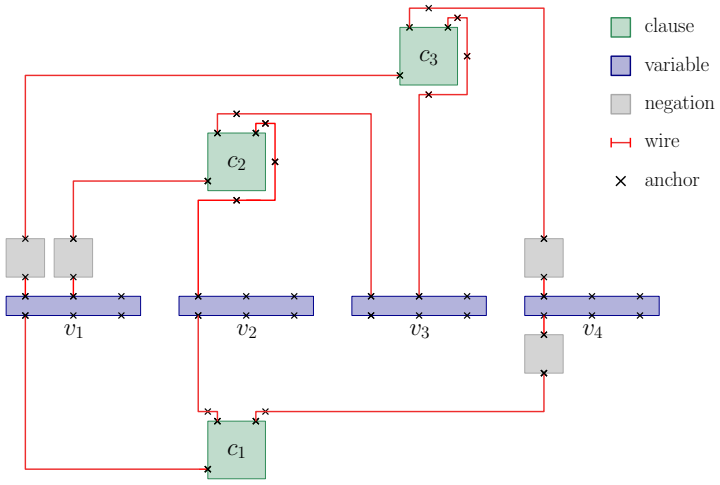

Figure 4.14: The triangulation instance corresponding to the planar 3SAT instance with clauses $c_1 = \overline{v_1} \vee v_2 \vee v_4$, $c_2 = v_1 \vee \overline{v_2} \vee v_3$ and $c_3 = v_1 \vee \overline{v_3} \vee \overline{v_4}$. The anchor points at which we connect two gadgets are depicted as crosses. Notice that some of the anchor points at the variable gadgets may be left unused.

We follow the notation in the construction of the clause gadgets and denote the anchors of a fixed clause gadget by a_1, a_2, a_3 (see Figure 4.10). If $G(c)$ lies above the horizontal axis, we connect the anchor point a_i to an anchor of the variable gadget at $G(v_i)$ for all $i \in \{1, 2, 3\}$. If $G(c)$ lies below the horizontal axis we connect a_1 to an anchor of the variable gadget at $G(v_1)$, a_2 to an anchor of the variable gadget at $G(v_3)$ and a_3 to an anchor of the variable gadget at $G(v_2)$. To connect the clause gadget at $G(c)$ at an anchor point a_i with an anchor point of the variable gadget at $G(v_j)$ we use a combination of wire gadgets. Notice that it is possible to connect all clause gadgets to the corresponding variable gadgets such that the wire gadgets of two distinct connections do not overlap (this is possible for a sufficiently large scaling factor $\gamma \in O(k)$, because the embedding is planar and rectilinear).

However, if the variable gadget of a variable that appears negated in the clause is connected to the a_3-anchor of the clause gadget, or if the variable gadget of a variable that appears non-negated in a clause is connected to the a_1- or a_2-anchor of the clause gadget, then we do not connect the clause gadget directly to the variable gadget, but we insert a negation gadget: If $G(c)$ lies above the horizontal axis we use a combination of wire gadgets to connect the anchor of the clause gadget to the a'-anchor of the negation gadget, and a wire segment to connect the a-anchor of the negation gadget to an anchor of the variable gadget. If $G(c)$ lies below the horizontal axis we use a combination of wire gadgets to connect the anchor of the clause gadget to the a-anchor of the negation gadget, and a wire segment to connect the a'-anchor of the negation gadget to an anchor of the variable gadget. If we choose the distance α between multiplier segments in a variable gadget to be at least 200 this construction can be done without the negation gadgets overlapping each other. Figure 4.14 shows the structure of the zero-error triangulation instance corresponding to our initial example.

Let $\mathcal{S}$ be the set of triangulation points and $\mathcal{R}$ the set of reference points of $\mathcal{I}_{\text{err}}$. Notice that $\mathcal{S}$ is contained in $\mathbb{Z}^2$ by construction. Furthermore we want to establish the property that $\mathcal{S}$ does not contain any forbidden points. This is already true for each of the discussed gadgets. Remember that we scaled the rectilinear embedding of $\mathcal{I}$ by

a factor $\gamma \in O(k)$ (the factor comes from the width of the variable gadget, which is in $O(k)$). If we pick γ sufficiently large (e.g., $\gamma = 1000k$) the gadgets do not overlap (excluding the overlap that occurs when two gadgets are combined with each other, which is explicitly allowed). Thus the instance does not contain forbidden triangulation points.

We are now able to prove the hardness of the zero-error triangulation problem.

Theorem 4.3.1. *The zero-error triangulation problem is* NP*-hard.*

Proof. Let $\mathcal{I}$ be an instance of the planar 3SAT problem and let $\mathcal{I}_{\text{err}}$ denote the corresponding instance of the zero-error triangulation problem. We have to guarantee that every zero-error triangulation D of the triangulation instance corresponds to a feasible assignment on the variables V and vice versa.

Suppose that there exists an assignment of the variables under which all clauses of the 3SAT instance are satisfied, and fix such an assignment. For every reference point that replaces a mandatory edge $\overline{st}$ we add $\overline{st}$ to the triangulation D. By Corollary 4.6.1 the error of D at such reference points is zero. For the reference points in the variable gadget at $G(v)$ we choose the triangulation D such that it is positive if the value of $v \in V$ is *true* and negative if the value of v is *false* in the assignment. Observe that by Lemma 4.4.3 a negation/wire gadget can be triangulated with zero error with a fixed signal at one of its anchor points. This zero-error triangulation of the negation/wire gadget has the negated/same signal on the remaining anchor points. We extend D on the negation/wire gadgets following the above observation. Now consider a clause gadget at $G(c)$ for some $c \in K$ and its three anchor points a_1, a_2, a_3 whose signals in D are already determined by the wire gadgets connected to them. As clause c is satisfied under the assignment, one of a_1, a_2 has a negative signal or a_3 has a positive signal. Thus at least one of the triangles triangulating $r = G(c) + (0, 11)$ with zero error can be added to D. In conclusion we see that D has zero error on this gadget.

Now suppose that there is a triangulation D of $\mathcal{S}$ with zero error. First observe that the mandatory edges must belong to D by Corollary 4.6.1. For $v \in V$ the triangulation must be either positive or negative on the variable gadget at $G(v)$ by Lemma 4.5.5. We assign to v the value *true* if D is positive on the variable gadget at $G(v)$ and *false* if it is negative. On all wire gadgets directly connected to a variable gadget, the triangulation must be either positive or negative by Lemma 4.5.5. If the triangulation is positive on a variable gadget, then it must be negative on all wire gadgets connected to it through a negation gadget and vice versa by Lemma 4.5.7. Lemma 4.5.6 then guarantees that all clauses of the 3SAT formula are satisfied under this assignment since all clause gadgets are triangulated with zero error.

It is left to show that the reduction works in polynomial time. The 3SAT instance can be embedded in polynomial time on an integral grid of size $O(k) \times O(k)$ [62]. Scaling the embedding by $\gamma \in O(k)$ and constructing the set of triangulation points $\mathcal{S} \subset \mathbb{Z}^2$ can be done in polynomial time. The same holds for the computation of $f(p_1, p_2) = p_1^2 + p_2^2$, as all triangulation points are integral. For the reference points and reference values we consider a reference point $r = (r_1, r_2) \in \mathcal{R}$ and its coupled circle C_r centered at a point $x = (x_1, x_2)$ with radius ρ. Recall that

$$h(r) = h_{C_r}(r) = 2x_1r_1 + 2x_2r_2 - x_1^2 - x_2^2 + \rho^2.$$

Thus if r and x are integral and the squared radius is an integer which is polynomial in k, then $h_{C_r}(r)$ is an integer polynomial in k. However there exist reference points and

reference values in our construction which are not integral. Observe that it is enough to show that there is a number $b \in \mathbb{N}_{>0}$ which is polynomial in the input size such that for every $r \in \mathcal{R}$ both br and $bh(r)$ are integral.

We first claim that this is true for the reference points if we set $b = 10$. Remember that a reference point which replaces a mandatory edge $e = \overline{st}$ for some $s, t \in \mathcal{S}$ is of the form $r_e = \frac{s+t}{2}$. Since s and t are both integral br_e is integral as well. All reference points of wire segments, wire gadgets and variable gadgets which do not replace mandatory edges are integral. For the clause gadget c_x at a point $x \in \mathbb{Z}^2$ we borrow the notation from the construction of the clause gadget. Observe that all points from $\mathcal{R}(c_x)\backslash\{r_1, r_2, r_3\}$ that do not replace mandatory edges are integral. Furthermore $5 \cdot r_i$ is integral for all $i = 1, 2, 3$, thus the claim is true for the clause gadget. Finally the negation gadget is a combination of simplified clause gadgets, wire segments, wire gadgets and multiplier gadgets, so the claim is also true for all reference points of the negation gadget.

For the reference values we first introduce an equivalence relation $\sim$ on the set of circles in $\mathbb{R}^2$ by saying that two circles C and C' are equivalent if there exists $v \in \mathbb{Z}^2$ such that $C' = v + C = \{v + c \mid c \in C\}$. Let $M = \{C_r \mid r \in \mathcal{R}\}$ be the set of all circles coupled to reference points in $\mathcal{R}$. Let $[C]$ be an equivalence class in the quotient set $M/\!\sim$ and suppose there exists a $b_C \in \mathbb{N}_{>0}$ such that $b_C x \in \mathbb{Z}^2$ and $b_C \rho^2 \in \mathbb{Z}$, where x is the center and ρ the radius of C. Then for all $C' \in [C]$ we also have $b_C x' \in \mathbb{Z}^2$ and $b_C \rho'^2 \in \mathbb{Z}$ for the center x' and the radius ρ' of C'. Since the cardinality of $M/\!\sim$ is constant it is left to observe that for every equivalence class $[C]$ there exists such a $b_C \in \mathbb{N}_{>0}$ which is polynomial in the input size. This property is clearly satisfied if C is centered at a reference point which replaces a mandatory edge. In the remaining cases C must contain at least three points $s = (s_1, s_2), t = (t_1, t_2), u = (u_1, u_2)$ lying on an integral grid of size $O(k^2) \times O(k^2)$. The center $x = (x_1, x_2)$ and the squared radius ρ^2 can be computed by solving the linear equation

$$\begin{pmatrix} 2s_1 & 2s_2 & 1 \\ 2t_1 & 2t_2 & 1 \\ 2u_1 & 2u_2 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \rho^2 \end{pmatrix} = \begin{pmatrix} s_1^2 + s_2^2 \\ t_1^2 + t_2^2 \\ u_1^2 + u_2^2 \end{pmatrix}.$$

Thus there is a $b_C \in \mathbb{N}_{>0}$ polynomial in k, such that $b_C x$ and $b_C \rho^2$ are integral. In conclusion we see that there exists a number $b \in \mathbb{N}_{>0}$ which is polynomial in the input size such that $bh(r)$ is integral for all $r \in \mathcal{R}$. Thus the reduction can be performed in polynomial time. □

Corollary 4.3.2. *The minimum-error triangulation problem cannot be approximated within any multiplicative factor in polynomial time unless* P = NP.

Proof. Every polynomial time approximation algorithm to the minimum-error triangulation problem yields a polynomial time algorithm to the zero-error triangulation problem. As the zero-error triangulation problem is NP-hard by Theorem 4.3.1 such a polynomial time approximation algorithm does not exist unless P = NP. □

4.8 The Paraboloid

It is left to prove the claimed properties of the unit paraboloid. The graph of the function $f\colon \mathbb{R}^2 \to \mathbb{R}$ with $f(p_1, p_2) = p_1^2 + p_2^2$ is the paraboloid

$$\Gamma_f = \{(p_1, p_2, p_1^2 + p_2^2) \mid (p_1, p_2) \in \mathbb{R}^2\}.$$

Remember that a circle C with radius ρ around $x = (x_1, x_2) \in \mathbb{R}^2$ defines the function $h_C(r) = 2x_1r_1 + 2x_2r_2 - x_1^2 - x_2^2 + \rho^2$ for $r = (r_1, r_2) \in \mathbb{R}^2$. Let

$$\Gamma_C = \{(r_1, r_2, h_C(r)) \mid r = (r_1, r_2) \in \mathbb{R}^2\}$$

denote the graph of h_C. We first repeat some useful properties of f and h_C.

Lemma 4.8.1. *For $y \in \mathbb{R}^2$ we have $f(y) - h_C(y) = \|y - x\|_2^2 - \rho^2$.*

Proof. We have

$$\begin{aligned} f(y) - h_C(y) &= y_1^2 + y_2^2 - 2x_1y_1 - 2x_2y_2 + x_1^2 + x_2^2 - \rho^2 \\ &= (y_1 - x_1)^2 + (y_2 - x_2)^2 - \rho^2 \\ &= \|y - x\|_2^2 - \rho^2. \end{aligned}$$

□

Lemma 4.8.2. *We have $\Gamma_C \cap \Gamma_f = \{(y_1, y_2, y_1^2 + y_2^2) \mid (y_1, y_2) \in C\}$.*

Proof. We have

$$\begin{aligned} \{(y_1, y_2, y_1^2 + y_2^2) \mid (y_1, y_2) \in C\} &= \{(y_1, y_2, y_1^2 + y_2^2) \mid (y_1 - x_1)^2 + (y_2 - x_2)^2 = \rho^2\} \\ &= \{(y_1, y_2, y_3) \mid y_3 - 2x_1y_1 - 2x_2y_2 + x_1^2 + x_2^2 - \rho^2 = 0, y_3 = y_1^2 + y_2^2\} \\ &= \Gamma_C \cap \Gamma_f. \end{aligned}$$

□

We are now able to prove Lemmas 4.4.3 and 4.4.4.

Lemma 4.4.3. *Let r be a point of $\mathcal{R}$ and let $T \subset \mathbb{R}^2$ be a triangle with vertices $s, t, u \in \mathcal{S}$ and $r \in \mathsf{conv}(\{s, t, u\} \cap C_r)$. Then r is represented with zero error by T.*

Proof. By Lemma 4.8.2 we know that $f(v) = h_{C_r}(v)$ for all $v \in \{s, t, u\} \cap C_r$. As h_{C_r} is affine this means that r is represented with zero error by T. □

Lemma 4.4.4. *Let $T \subset \mathbb{R}^2$ be a triangle with vertices $s, t, u \in \mathcal{S}$ representing $r \in \mathcal{R}$ with zero error. If $r \notin \mathsf{conv}(\{s, t, u\} \cap C_r)$, then $\{s, t, u\}$ has a non-empty intersection with I_{C_r} and O_{C_r}.*

Proof. We pick a convex combination $\lambda s + \mu t + \gamma u = r$. As T represents r with zero error, we have

$$h_{C_r}(r) = \lambda f(s) + \mu f(t) + \gamma f(u).$$

Since h_{C_r} is affine also

$$h_{C_r}(r) = h_{C_r}(\lambda s + \mu t + \gamma u) = \lambda h_{C_r}(s) + \mu h_{C_r}(t) + \gamma h_{C_r}(u).$$

Combining the two equations we obtain

$$0 = \lambda(f(s) - h_{C_r}(s)) + \mu(f(t) - h_{C_r}(t)) + \gamma(f(u) - h_{C_r}(u)).$$

Observe that $f(p) - h_{C_r}(p) < 0$ for $p \in I_{C_r}$ and $f(p) - h_{C_r}(p) > 0$ for $p \in O_{C_r}$ by Lemma 4.8.1. We distinguish two cases: If points in $\{s, t, u\} \cap I_{C_r}$ and $\{s, t, u\} \cap O_{C_r}$ appear with factor zero in the above equation, then $r \in \mathsf{conv}(\{s, t, u\} \cap C_r)$ contradicting our assumption. Otherwise the sets $\{s, t, u\} \cap I_{C_r}$ and $\{s, t, u\} \cap O_{C_r}$ must be non-empty. □

We need some additional statement about the behavior of the error under orthogonal transformations and translations of the triangle and the reference point it is representing.

Lemma 4.5.1. *Let $T \subset \mathbb{R}^2$ be a triangle representing $r \in \mathcal{R}$ with error ϵ. Let $g\colon \mathbb{R}^2 \to \mathbb{R}^2$ be an orthogonal transformation or translation. Let $\overline{T} = g(T), \overline{r} = g(r)$ and $\overline{C}_r = g(C_r)$. Then $\overline{T}$ is a triangle containing $\overline{r}$, $\overline{C}_r$ is a circle containing $\overline{r}$ in its interior and $(f_{\overline{T}}(\overline{r}) - h_{\overline{C}_r}(\overline{r}))^2 = \epsilon$.*

Proof. Remember that an orthogonal transformation in $\mathbb{R}^2$ is a rotation or a reflection. Thus for every possible choice of g we have that $\overline{T}$ is a triangle containing $\overline{r}$ and $\overline{C}_r$ is a circle containing $\overline{r}$ in its interior. Therefore the error of $\overline{T}$ at $\overline{r}$ given by $(f_{\overline{T}}(\overline{r}) - h_{\overline{C}_r}(\overline{r}))^2$ is properly defined. Let s, t, u be the vertices of T. We pick a convex combination $\lambda s + \mu t + \gamma u = r$ and obtain

$$\begin{aligned}\epsilon &= (\lambda f(s) + \mu f(t) + \gamma f(u) - h_{C_r}(r))^2 \\ &= (\lambda(f(s) - h_{C_r}(s)) + \mu(f(t) - h_{C_r}(t)) + \gamma(f(u) - h_{C_r}(u)))^2.\end{aligned}$$

By Lemma 4.8.1 the last part depends on the radius of C_r and the distance between its center and s, t, u. These values do not change after applying g on T, r and C_r. Thus we obtain

$$\begin{aligned}\epsilon &= \Big(\lambda\Big(f(g(s)) - h_{\overline{C}_r}(g(s))\Big) + \mu\Big(f(g(t)) - h_{\overline{C}_r}(g(t))\Big) + \gamma\Big(f(g(u)) - h_{\overline{C}_r}(g(u))\Big)\Big)^2 \\ &= \Big(\lambda f(g(s)) + \mu f(g(t)) + \gamma f(g(u)) - \lambda h_{\overline{C}_r}(g(s)) - \mu h_{\overline{C}_r}(g(t)) - \gamma h_{\overline{C}_r}(g(u))\Big)^2 \\ &= (f_{\overline{T}}(\overline{r}) - h_{\overline{C}_r}(\overline{r}))^2.\end{aligned}$$

□

Chapter 5

Conclusion

In this thesis we first discussed two well-known problems from the area of clustering, clustering with lower bounds and hierarchical clustering. Furthermore we considered the complexity of the minimum-error triangulation problem. In the following we give a brief summary of the results of this thesis and discuss further research questions.

In Chapter 2 we study clustering with lower bounds. We show how to transform a solution for facility location with lower bounds into a k-clustering with lower bounds. This approach is independent of the algorithm that computes the facility location solution, so an improvement of the approximation factor for facility location directly implies an improvement of the approximation factor for its k-clustering variant. Furthermore we consider a relaxation of lower bounds called weak lower bounds and show that it is possible to obtain a $(13+\epsilon)$-approximation for k-median with 2-weak lower bounds. Our insights on weak lower bounds also yield the first bi-criteria algorithm for k-means with lower bounds. So far there do not exist approximation algorithms for k-means with lower bounds. It would be interesting to know whether the algorithms presented in [72, 5] for facility location with lower bounds can be adapted to work for k-means with lower bounds. Since the first step of the adapted algorithms would involve finding a bi-criteria solution for k-means with lower bounds, which does exist, as we show in this thesis, we assume that the remaining steps may also be adapted. It is also reasonable to ask whether there is an improvement of the approximation factors. So far the best approximation factor for facility location with lower bounds is 82.6 for uniform lower bounds [5] and 4000 for non-uniform lower bounds [61]. As we show they imply algorithms for k-median with lower bounds with approximation factor 168 for uniform lower bounds and 12006 for non-uniform lower bounds, which are rather high.

In Chapter 3 we studied hierarchical clustering. For a fixed objective the approximation factor of a hierarchical clustering is computed by comparing the cost of its k-clustering to the cost of an optimal k-clustering for every possible k. The price of hierarchy ρ_{cost} now describes the smallest possible approximation factor for a hierarchical clustering with respect to the objective cost. We show that the price of hierarchy for drad equals 4 and for $\mathsf{rad}, \mathsf{diam}$ it equals $3+2\sqrt{2}$. However this does not imply that there exist polynomial time algorithms with these approximation guarantees. The currently best polynomial time approximation algorithm achieves a guarantee of 8 [39, 27]. So it is still an open question whether it is NP-hard to compute hierarchical clusterings with approximation factor $\alpha < 8$.

We assume that our construction of the instance for the lower bound on ρ_{cost} could be adapted to prove lower bounds on ρ_{cost} for other objective functions as med and

mean. So far we do not now any non-trivial lower bounds on ρ_{cost} for these objectives. In general it might be interesting to analyze the price of hierarchy for other objective functions.

In addition to the analysis of the price of hierarchy, we studied one popular algorithm for hierarchical clustering, called complete linkage. Our results show that the approximation factor of complete linkage in a general metric space is in $\Omega(k)$ for drad and diam. For drad we are able to prove a matching upper bound of $O(k)$. For diam we are only able to prove an upper bound of $O(k^{\ln(3)/\ln(2)})$. Thus it is still open whether the approximation guarantee of complete linkage for diam is in $\Theta(k)$. We know that in a Euclidean space of constant dimension complete linkage computes an $O(1)$-approximation [1, 44]. However the respective factor is exponential or doubly exponential in the dimension for rad or diam and it is still open if one can improve upon these dependencies. One other popular agglomerative clustering algorithm is average linkage, which merges the two clusters with smallest average distance to each other in every step. It is not known how average linkage performs with respect to the objective functions drad, diam. In particular it would be interesting to know whether one can adapt the lower bound instance for complete linkage to obtain lower bounds on the approximation guarantee of average linkage.

In Chapter 4 we studied the complexity of the minimum-error triangulation problem. We showed that it is NP-hard to decide whether there exists a triangulation with zero error. This implies that the minimum-error triangulation problem cannot be approximated within any factor. Furthermore it implies the inapproximability of the following generalization: minimizing the distance between f_D and h on $\mathcal{R}$ for any metric on $\mathbb{R}^m$, especially the L_p-metric and the L_∞-metric.

Bibliography

[1] Marcel R. Ackermann, Johannes Blömer, Daniel Kuntze, and Christian Sohler. Analysis of agglomerative clustering. *Algorithmica*, 69(1):184–215, 2014. doi:10.1007/s00453-012-9717-4.

[2] Pankaj K. Agarwal and Subhash Suri. Surface approximation and geometric partitions. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '94, pages 24–33, USA, 1994. Society for Industrial and Applied Mathematics.

[3] Gagan Aggarwal, Rina Panigrahy, Tomás Feder, Dilys Thomas, Krishnaram Kenthapadi, Samir Khuller, and An Zhu. Achieving anonymity via clustering. *ACM Transactions on Algorithms (TALG)*, 6(3):49:1–49:19, 2010. doi:10.1145/1798596.1798602.

[4] Sara Ahmadian and Chaitanya Swamy. Improved approximation guarantees for lower-bounded facility location. In *Proceedings of the 10th International Workshop on Approximation and Online Algorithms (WAOA)*, pages 257–271, 2012. doi:10.1007/978-3-642-38016-7_21.

[5] Sara Ahmadian and Chaitanya Swamy. Approximation algorithms for clustering problems with lower bounds and outliers. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 69:1–69:15, 2016. doi:10.4230/LIPIcs.ICALP.2016.69.

[6] Lyuba Alboul, Gertjan Kloosterman, Cornelis Traas, and Ruud van Damme. Best data-dependent triangulations. *Journal of Computational and Applied Mathematics*, 119(1):1–12, 2000. doi:10.1016/S0377-0427(00)00368-X.

[7] Efthymios Anagnostou and Derek Corneil. Polynomial-time instances of the minimum weight triangulation problem. *Computational Geometry*, 3(5):247 – 259, 1993. URL: http://www.sciencedirect.com/science/article/pii/092577219390016Y, doi:https://doi.org/10.1016/0925-7721(93)90016-Y.

[8] David Arthur and Sergei Vassilvitskii. How slow is the k-means method? In *Proceedings of the 22nd ACM Symposium on Computational Geometry (SoCG), 2006*, pages 144–153, 2006. doi:10.1145/1137856.1137880.

[9] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, 2007. URL: http://dl.acm.org/citation.cfm?id=1283383.1283494.

[10] Anna Arutyunova. On variants of lower-bounded facility location. Master's thesis, University of Bonn, 2019.

[11] Anna Arutyunova, Anne Driemel, Jan-Henrik Haunert, Herman J. Haverkort, Jürgen Kusche, Elmar Langetepe, Philip Mayer, Petra Mutzel, and Heiko Röglin. Minimum-error triangulations for sea surface reconstruction. In *38th International Symposium on Computational Geometry (SoCG)*, pages 7:1–7:18, 2022. doi:10.4230/LIPIcs.SoCG.2022.7.

[12] Anna Arutyunova, Anne Driemel, Jan-Henrik Haunert, Herman J. Haverkort, Jürgen Kusche, Elmar Langetepe, Philip Mayer, Petra Mutzel, and Heiko Röglin. Minimum-error triangulations for sea surface reconstruction. *CoRR*, 2022. arXiv:2203.07325, doi:10.48550/arXiv.2203.07325.

[13] Anna Arutyunova, Anna Großwendt, Heiko Röglin, Melanie Schmidt, and Julian Wargalla. Upper and lower bounds for complete linkage in general metric spaces. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 18:1–18:22, 2021. doi:10.4230/LIPIcs.APPROX/RANDOM.2021.18.

[14] Anna Arutyunova and Heiko Röglin. The price of hierarchical clustering. In *30th Annual European Symposium on Algorithms (ESA)*, volume 244, pages 10:1–10:14, 2022. doi:10.4230/LIPIcs.ESA.2022.10.

[15] Anna Arutyunova and Heiko Röglin. The price of hierarchical clustering. *CoRR*, abs/2205.01417, 2022. arXiv:2205.01417, doi:10.48550/arXiv.2205.01417.

[16] Anna Arutyunova and Melanie Schmidt. Achieving anonymity via weak lower bound constraints for k-median and k-means. *CoRR*, 2020. arXiv:2009.03078, doi:10.48550/arXiv.2009.03078.

[17] Anna Arutyunova and Melanie Schmidt. Achieving anonymity via weak lower bound constraints for k-median and k-means. In *38th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 7:1–7:17, 2021. doi:10.4230/LIPIcs.STACS.2021.7.

[18] Sayan Bandyapadhyay, Fedor V. Fomin, and Kirill Simonov. On coresets for fair clustering in metric and euclidean spaces and their applications. In *48th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 23:1–23:15, 2021. doi:10.4230/LIPIcs.ICALP.2021.23.

[19] Manisha Bansal, Naveen Garg, and Neelima Gupta. A 5-approximation for capacitated facility location. In Leah Epstein and Paolo Ferragina, editors, *Algorithms - ESA 2012 - 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings*, volume 7501 of *Lecture Notes in Computer Science*, pages 133–144. Springer, 2012. doi:10.1007/978-3-642-33090-2_13.

[20] Suman Kalyan Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 4955–4966, 2019. URL: https://proceedings.neurips.cc/paper/2019/hash/fc192b0c0d270dbf41870a63a8c76c2f-Abstract.html.

[21] Marshall Bern and David Eppstein. Mesh generation and optimal triangulation. In *Computing in Euclidean Geometry*, 1992. doi:10.1142/9789814355858_0002.

[22] Felix Bock. Hierarchy cost of hierarchical clusterings. *Journal of Combinatorial Optimization*, 2022. doi:10.1007/s10878-022-00851-4.

[23] Magdalene Borgelt, Christian Borgelt, and Christos Levcopoulos. Fixed parameter algorithms for the minimum weight triangulation problem. *Int. J. Comput. Geometry Appl.*, 18:185–220, 06 2008. doi:10.1142/S0218195908002581.

[24] Jeffrey L. Brown. Vertex based data dependent triangulations. *Computer Aided Geometric Design*, 8(3):239–251, 1991. doi:10.1016/0167-8396(91)90008-Y.

[25] Jaroslaw Byrka, Thomas W. Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Trans. Algorithms*, 13(2):23:1–23:31, 2017. doi:10.1145/2981561.

[26] Moses Charikar and Vaggos Chatziafratis. Approximate hierarchical clustering via sparsest cut and spreading metrics. In *Proc. of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 841–854, 2017. doi:10.1137/1.9781611974782.53.

[27] Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. *SIAM J. Comput.*, 33(6):1417–1440, 2004. doi:10.1137/S0097539702418498.

[28] Moses Charikar and Shi Li. A dependent lp-rounding approach for the k-median problem. In *Automata, Languages, and Programming - 39th International Colloquium (ICALP)*, pages 194–205, 2012. doi:10.1007/978-3-642-31594-7_17.

[29] Siu-Wing Cheng, Mordecai J. Golin, and Jeffrey Tsang. Expected case analysis of {221}-skeletons with applications to the construction of minimum-weight triangulations. Master's thesis, Hong Kong University of Science and Technology, 1995.

[30] John A. Church, Neil J. White, Richard Coleman, Kurt Lambeck, and Jerry X. Mitrovica. Estimates of the Regional Distribution of Sea Level Rise over the 1950–2000 Period. *Journal of Climate*, 17(13):2609–2625, July 2004. doi:10.1175/1520-0442(2004)017<2609:EOTRDO>2.0.CO;2.

[31] Vincent Cohen-Addad, Hossein Esfandiari, Vahab S. Mirrokni, and Shyam Narayanan. Improved approximations for euclidean k-means and k-median, via nested quasi-independent sets. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1621–1628. ACM, 2022. doi:10.1145/3519935.3520011.

[32] Vincent Cohen-Addad, Fabrizio Grandoni, Euiwoong Lee, and Chris Schwiegelshohn. Breaching the 2 LMP approximation barrier for facility location with applications to k-median. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 940–986. SIAM, 2023. doi:10.1137/1.9781611977554.ch37.

[33] Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. Hierarchical clustering: Objective functions and algorithms. In *Proc. of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 378–397, 2018. doi:10.1137/1.9781611975031.26.

[34] Vincent Cohen-Addad and Karthik C. S. Inapproximability of clustering in lp metrics. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 519–539. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00040.

[35] Stephen A. Cook. The complexity of theorem-proving procedures (1971). In *Ideas that created the future. Classic papers of computer science*, pages 333–338. Cambridge, MA: MIT Press, 2021.

[36] Wenqiang Dai. A 16-competitive algorithm for hierarchical median problem. *SCIENCE CHINA Information Sciences*, 57(3):1–7, 2014. doi:10.1007/s11432-014-5065-0.

[37] Aparna Das and Claire Kenyon-Mathieu. On hierarchical diameter-clustering and the supplier problem. *Theory Comput. Syst.*, 45(3):497–511, 2009. doi:10.1007/s00224-009-9186-6.

[38] Sanjoy Dasgupta. A cost function for similarity-based hierarchical clustering. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 118–127, 2016. doi:10.1145/2897518.2897527.

[39] Sanjoy Dasgupta and Philip M. Long. Performance guarantees for hierarchical clustering. *Journal of Computer and System Sciences*, 70(4):555–569, 2005. doi:10.1016/j.jcss.2004.10.006.

[40] Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational geometry: algorithms and applications, 3rd Edition*. Springer, 2008. URL: https://www.worldcat.org/oclc/227584184.

[41] Nira Dyn, David Levin, and Samuel Rippa. Data Dependent Triangulations for Piecewise Linear Interpolation. *IMA Journal of Numerical Analysis*, 10(1):137–154, 01 1990. arXiv:https://academic.oup.com/imajna/article-pdf/10/1/137/1956877/10-1-137.pdf, doi:10.1093/imanum/10.1.137.

[42] P. Gilbert. New results on planar triangulations. Master's thesis, University of Illinois, Coordinated Science Lab, Urbana, IL, USA, 1979.

[43] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science (TCS)*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.

[44] Anna Großwendt and Heiko Röglin. Improved analysis of complete-linkage clustering. *Algorithmica*, 78(4):1131–1150, 2017. doi:10.1007/s00453-017-0284-6.

[45] Anna Großwendt, Heiko Röglin, and Melanie Schmidt. Analysis of ward's method. In *Proc. of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2939–2957, 2019. doi:10.1137/1.9781611975482.182.

[46] Anna-Klara Großwendt. *Theoretical Analysis of Hierarchical Clustering and the Shadow Vertex Algorithm*. PhD thesis, University of Bonn, 2020. URL: http://hdl.handle.net/20.500.11811/8348.

[47] Joachim Gudmundsson, Mikael Hammar, and Marc van Kreveld. Higher order Delaunay triangulations. *Computational Geometry*, 23(1):85 – 98, 2002. URL: http://www.sciencedirect.com/science/article/pii/S092577210100027X, doi:https://doi.org/10.1016/S0925-7721(01)00027-X.

[48] Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms*, 31(1):228–248, 1999. doi:10.1006/jagm.1998.0993.

[49] Sudipto Guha, Adam Meyerson, and Kamesh Munagala. Hierarchical placement and network design problems. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 603–612, 2000. doi:10.1109/SFCS.2000.892328.

[50] Lu Han, Chunlin Hao, Chenchen Wu, and Zhenning Zhang. Approximation algorithms for the lower-bounded k-median and its generalizations. In *Computing and Combinatorics - 26th International Conference (COCOON), Proceedings*, volume 12273 of *Lecture Notes in Computer Science*, pages 627–639. Springer, 2020. doi:10.1007/978-3-030-58150-3_51.

[51] Lu Han, Chunlin Hao, Chenchen Wu, and Zhenning Zhang. Approximation algorithms for the lower-bounded knapsack median problem. In *Algorithmic Aspects in Information and Management - 14th International Conference (AAIM), Proceedings*, pages 119–130, 2020. doi:10.1007/978-3-030-57602-8_11.

[52] Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM*, 33(3):533–550, 1986. doi:10.1145/5925.5933.

[53] Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM*, 48(2):274–296, 2001. doi:10.1145/375827.375845.

[54] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. *Computational Geometry*, 28(2-3):89–112, 2004.

[55] David R. Karger and Maria Minkoff. Building Steiner trees with incomplete global knowledge. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 613–623, 2000. doi:10.1109/SFCS.2000.892329.

[56] G.T. Klincsek. Minimal triangulations of polygonal domains. In Peter L. Hammer, editor, *Combinatorics 79*, volume 9 of *Annals of Discrete Mathematics*, pages 121–123. Elsevier, 1980. URL: https://www.sciencedirect.com/science/article/pii/S016750600870044X, doi:https://doi.org/10.1016/S0167-5060(08)70044-X.

[57] Donald E. Knuth and Arvind Raghunathan. The problem of compatible representatives. *SIAM Journal on Discrete Mathematics*, 5(3):422–427, 1992. doi:10.1137/0405033.

[58] Charles L. Lawson. Software for C^1 surface interpolation. In John R. Rice, editor, *Mathematical Software*, pages 161–194. Academic Press, 1977. doi:10.1016/B978-0-12-587260-7.50011-X.

[59] Christos Levcopoulos and Drago Krznaric. Quasi-greedy triangulations approximating the minimum weight triangulation. *J. Algorithms*, 27(2):303–338, 1998.

[60] Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Information and Computation*, 222:45–58, 2013. doi:10.1016/j.ic.2012.01.007.

[61] Shi Li. On facility location with general lower bounds. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2279–2290, 2019. doi:10.1137/1.9781611975482.138.

[62] David Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11:329–343, 1982. doi:10.1137/0211025.

[63] Guolong Lin, Chandrashekhar Nagarajan, Rajmohan Rajaraman, and David P. Williamson. A general approach for incremental approximation and hierarchical clustering. *SIAM Journal on Computing*, 39(8):3633–3669, 2010. doi:10.1137/070698257.

[64] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. doi:10.1109/TIT.1982.1056489.

[65] Jirí Matousek. On approximate geometric k-clustering. *Discret. Comput. Geom.*, 24(1):61–84, 2000. doi:10.1007/s004540010019.

[66] Sakib A. Mondal. An improved approximation algorithm for hierarchical clustering. *Pattern Recognit. Lett.*, 104:23–28, 2018. doi:10.1016/j.patrec.2018.01.015.

[67] Wolfgang Mulzer and Günter Rote. Minimum-weight triangulation is NP-hard. *Journal of the ACM*, 55(2):1–29, May 2008. URL: http://dx.doi.org/10.1145/1346330.1346336, doi:10.1145/1346330.1346336.

[68] Alina Nitzke, Benjamin Niedermann, Luciana Fenoglio-Marc, Jürgen Kusche, and Jan-Henrik Haunert. Reconstructing the dynamic sea surface from tide gauge records using optimal data-dependent triangulations. *Computers & Geosciences*, 157:104920, 2021. doi:10.1016/j.cageo.2021.104920.

[69] Marco Olivieri and Giorgio Spada. Spatial sea-level reconstruction in the Baltic Sea and in the Pacific Ocean from tide gauges observations. *Annals of Geophysics*, 59(3), 2016. URL: https://www.annalsofgeophysics.eu/index.php/annals/article/view/6966, doi:10.4401/ag-6966.

[70] C. Greg Plaxton. Approximation algorithms for hierarchical location problems. *Journal of Computer and System Sciences*, 72(3):425–443, 2006. doi:10.1016/j.jcss.2005.09.004.

[71] Rodrigo I. Silveira and Marc van Kreveld. Optimal higher order Delaunay triangulations of polygons. *Computational Geometry*, 42(8):803 – 813, 2009. Special Issue on the 23rd European Workshop on Computational Geometry. URL: `http://www.sciencedirect.com/science/article/pii/S0925772109000224`, `doi:https://doi.org/10.1016/j.comgeo.2008.02.006`.

[72] Zoya Svitkina. Lower-bounded facility location. *ACM Transactions on Algorithms (TALG)*, 6(4):69, 2010. `doi:10.1145/1824777.1824789`.

[73] Kai Wang, Chor-Pang Lo, George A. Brook, and Hamid R. Arabnia. Comparison of existing triangulation methods for regularly and irregularly spaced height fields. *International Journal of Geographical Information Science*, 15(8):743–762, 2001. `doi:10.1080/13658810110074492`.

[74] Yuyan Wang and Benjamin Moseley. An objective for hierarchical clustering in euclidean space and its connection to bisecting k-means. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6307–6314, 2020. `doi:10.1609/aaai.v34i04.6099`.

[75] Joe H. Ward, Jr. Hierarchical grouping to optimize an objective function. *J. of the Am. Stat. Assoc.*, 58:236–244, 1963. `doi:10.1080/01621459.1963.10500845`.