Upper and Lower Bounds for Complete Linkage in General Metric Spaces

Anna Arutyunova ✉
Universität Bonn, Germany

Anna Großwendt ✉
Universität Bonn, Germany

Heiko Röglin ✉
Universität Bonn, Germany

Melanie Schmidt ✉
Universität Köln, Germany

Julian Wargalla ✉
Universität Köln, Germany

Abstract

In a hierarchical clustering problem the task is to compute a series of mutually compatible clusterings of a finite metric space (P, dist). Starting with the clustering where every point forms its own cluster, one iteratively merges two clusters until only one cluster remains. Complete linkage is a well-known and popular algorithm to compute such clusterings: in every step it merges the two clusters whose union has the smallest radius (or diameter) among all currently possible merges. We prove that the radius (or diameter) of every k-clustering computed by complete linkage is at most by factor $O(k)$ (or $O(k^2)$) worse than an optimal k-clustering minimizing the radius (or diameter). Furthermore we give a negative answer to the question proposed by Dasgupta and Long [6], who show a lower bound of $\Omega(\log(k))$ and ask if the approximation guarantee is in fact $\Theta(\log(k))$. We present instances where complete linkage performs poorly in the sense that the k-clustering computed by complete linkage is off by a factor of $\Omega(k)$ from an optimal solution for radius and diameter. We conclude that in general metric spaces complete linkage does not perform asymptotically better than single linkage, merging the two clusters with smallest inter-cluster distance, for which we prove an approximation guarantee of $O(k)$.

2012 ACM Subject Classification Theory of computation → Facility location and clustering

Keywords and phrases Hierarchical Clustering, Complete Linkage, agglomerative Clustering, k-Center

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2021.18

Category APPROX

Funding This work has been supported by DFG grants RO 5439/1-1 and SCHM 2765/1-1.

1 Introduction

The *k-clustering* problem asks for a partition of a point set in a metric space into k subsets (or clusters). To measure whether the data is clustered well, one option is to pick a center for every cluster and compute the maximum distance between a point and the center of its cluster. This objective is to be minimized and is known as *k-center*. A problem which is independent of the choice of centers is the *k-diameter* problem, where we want to minimize the maximum distance between two points lying in the same cluster. Observe that k-center and k-diameter are related to each other in the sense that for a fixed set P the cost of an optimal k-diameter clustering on P is at most twice the cost of an optimal k-center clustering, which again costs at most as much as an optimal k-diameter clustering. There are other

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021).
Editors: Mary Wootters and Laura Sanità; Article No. 18; pp. 18:1–18:22
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

objectives to measure the quality of a clustering where every point contributes to the cost of the clustering, for example *k-median* and *k-means*. Here we want to minimize the cost which equals the sum over all (squared) distances between a point and the center of its cluster.

The k-center problem is NP-hard to approximate with factor $\alpha < 2$ [12, 14]. This bound is tight, as both Gonzalez [7] and Hochbaum and Shmoys [13] show. Gonzalez's 2-approximation algorithm is a simple, but elegant greedy approach. Starting with an arbitrary point $p_1 \in P$, one constructs an enumeration $P = \{p_1, \ldots, p_{|P|}\}$ by successively choosing as p_{i+1} a point from P whose minimum distance to any point from $\{p_1, \ldots, p_i\}$ is maximal. Assigning every point from P to its closest neighbor among $p_1, \ldots, p_k$ (the centers) yields a 2-approximation for the k-center problem for all $k = 1, \ldots, |P|$. One can prove that the resulting clustering is also a 2-approximation to k-diameter. Another greedy approach is the reverse greedy algorithm, which starts with all data points as centers and iteratively removes a center such that the objective stays as small as possible. Hershkowitz and Kehne [11] show that this algorithm computes an $\Theta(k)$-approximation. Observe that both greedy algorithms compute an *incremental clustering* where the centers of a k-clustering are also centers of an l-clustering if $l \geq k$.

Gonzalez's algorithm [7] allows to compute good clusterings, even if one does not previously know an appropriate value for k. However, even successive clusterings computed by Gonzalez's algorithm and reverse greedy can be radically different and so it can be difficult to compare them and select one that seems appropriate for the task.

Another greedy approach known as *complete linkage* starts with every point in its own cluster and consecutively merges two clusters whose union has the smallest radius (or diameter when considering the k-diameter objective) among all possible cluster pairs. If we proceed like this until only one cluster remains, we also obtain a k-clustering for any possible $1 \leq k \leq |P|$. However, this time, the resulting clusterings are also *hierarchically compatible*: for all $\ell \geq k$ the ℓ-clustering is a refinement of the k-clustering. This makes it easier to compare such clusterings with each other and to choose an appropriate k-clustering. Also, this additional hierarchical structure is interesting in and of itself. Famous examples include phylogenetic trees that represent the relationship between animal species in biology.

A series of such hierarchically compatible clusterings $\mathscr{C}_1, \ldots, \mathscr{C}_{|P|}$ (with $\mathscr{C}_k$ being a k-clustering for all k) forms a *hierarchical clustering*. Complete linkage is a common and popular bottom-up approach to compute these and can be generalized to fit any k-clustering objective, resulting in so called *agglomerative clustering* methods. For hierarchical k-means this is Wards method [16].

To evaluate a hierarchical clustering $\mathscr{C}_1, \ldots \mathscr{C}_{|P|}$ we refer to the underlying k-clusterings: it is an *α-approximation* if the cost of $\mathscr{C}_k$ is at most α times that of an optimal k-clustering for all $1 \leq k \leq |P|$.

Related work. For hierarchical k-center and k-diameter, constant factor approximations are known. For both problems Dasgupta and Long [6] and Charikar et al. [4] give a polynomial-time 8-approximation. In [15] Lin et al. introduce the concept of nesting. Using this technique, every approximation algorithm to a k-clustering objective that satisfies their nesting property can be converted into an algorithm for its hierarchical version. Especially k-median and k-means satisfy this property and thus (in combination with the currently best constant factor approximations for k-median [3] and k-means [2]), polynomial time constant factor approximations do indeed exist for the hierarchical k-median/k-means problem. Yet the resulting guarantees are relatively high (≈ 56 for k-median and ≈ 3662 for k-means). Nesting can also be applied to k-center/k-diameter but does not improve upon the 8-approximation.

As optimal k-clusterings are not necessarily hierarchically compatible, even assuming unlimited computation power 1-approximations do not exist in general. Das and Kenyon-Mathieu [5] give an instance for the diameter and Großwendt [10] for the radius where the best hierarchical clustering is a 2-approximation. Using the concept of nesting by Lin et al. [15], Großwendt [10] proves an existential upper bound of 4 for hierarchical k-center.

However, greedy algorithms are more common in practical applications. There exist several theoretical results on upper and lower bounds on the approximation factor for complete linkage. For metrics induced by norms in $\mathbb{R}^d$, especially the Euclidean metric, Ackermann et al. [1] prove that, assuming the dimension d to be constant, complete linkage computes for both the k-center and k-diameter objective an $O(\log(k))$ approximation. This was later improved by Großwendt and Röglin [8] to $O(1)$. Both works distinguish between two variants of k-center: one where centers must be from the set P and the second where they can be arbitrary points chosen from the whole space $\mathbb{R}^d$. In the first case the approximation factor shown in [1, 8] depends linearly on d and in the second case exponentially on d. For the k-diameter problem it even depends doubly exponentially on the dimension. Furthermore Ackermann et al. prove for the l_p-metric with $1 \leq p < \infty$ a lower bound of $\Omega(\sqrt[p]{\log(d)})$ for complete linkage for k-diameter and k-center with centers drawn from P [1].

Little is known about complete linkage in general metric spaces. Dasgupta and Long show in [6] that the lower bound is in $\Omega(\log(k))$. With an approach to upper bound the increase in cost by a complete linkage merge, which we borrow from [1], we obtain in a relatively straightforward manner an upper bound of $O(\log(|P| - k))$ for complete linkage for k-center.

There exist few results for agglomerative clustering regarding other objectives. Großwendt et al. [9] analyze Ward's method for k-means, and show that if the clusters of an optimal k-means clustering are sufficiently far apart, Ward's method computes a 2-approximation and under some additional assumptions in fact reconstructs the optimal clustering.

Our Results. We study upper and lower bounds for the complete linkage algorithm in general metric spaces for the k-center and k-diameter objective. For k-center in general metric spaces it is reasonable to assume that centers can be only drawn from P and thus we only consider this variant. Our main results are:

- A lower bound of $\Omega(k)$ for complete linkage for k-center and k-diameter, which improves the currently highest lower bound of $\Omega(\log(k))$ by Dasgupta and Long [6] significantly.
- An upper bound of $O(k)$ for k-center and an upper bound of $O(k^2)$ for k-diameter, which are to the best of the authors' knowledge the first non-trivial upper bounds for complete linkage in general metric spaces.

The lower bound $\Omega(k)$ is surprising as it shows that complete linkage does not perform asymptotically better than single linkage, which merges the two clusters with smallest distance to each other (the distance of two clusters is the smallest distance between two of their points). Dasgupta and Long [6] prove a lower bound of $\Omega(k)$ for single linkage and we show that the approximation factor is in fact $\Theta(k)$. As single linkage is not designed to minimize the radius or diameter of emerging clusters, it is a natural assumption that it performs worse than complete linkage. However our results show that this assumption is generally not true. It is even still open if complete linkage for k-diameter performs as good as single linkage, as we are only able to prove an upper bound of $O(k^2)$.

Techniques. One of the biggest and most well-known issues concerning single linkage is that of chaining. If there is a sequence of points $x_1, \dots, x_k \in P$ with $\operatorname{dist}(x_i, x_{i+1})$ relatively small for all i, then single linkage might merge all of them together, despite the resulting cluster

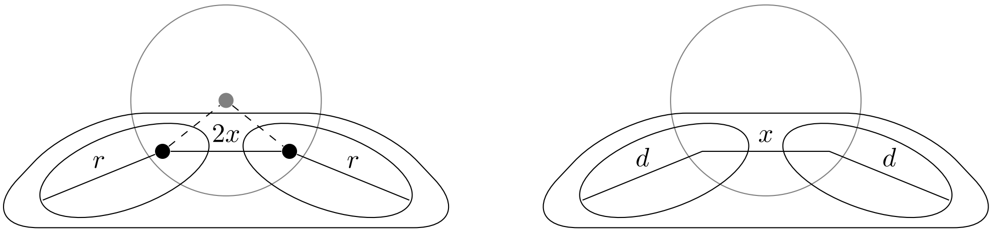

Figure 1 On the left we see two k-center clusters with radius r whose centers lie in the same optimal cluster. The radius of the merged cluster is at most $r + 2x$. On the right we have a similar situation for k-diameter but the merged cluster has diameter at most $2d + x$.

being quite large. Dasgupta and Long 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, $\mathrm{dist}((x,y),(x',y')) = \mathbf{1}_{x \neq x'} + \log_2(1 + |y - y'|)$. 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 of Ackermann et al. [1] one can show for complete linkage an upper bound of $\log(|P| - k)$ for k-center. This comes from the following easy property, which is true for the radius but cannot be transferred to diameter: Suppose the optimal k-center solution $\mathscr{O}$ 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 if we replace k-center by k-diameter we see that the cost is more than doubled in the worst case (see Figure 1), 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 $\mathscr{O}$ which it intersects. As $\mathscr{O}$ 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 $O(k^2)$ for k-diameter and, in combination with the $\log(|P| - k)$ upper bound, an upper bound of $O(k)$ for k-center.

2 Preliminaries

Let (P, dist) be a metric space with n points and $1 \leq k \leq n$. The *k-center* problem asks for a partition of P into k clusters $\mathscr{C} = \{C_1, \dots, C_k\}$. The cost of cluster C_i is given by $\mathrm{cost}(C_i) = \min_{c \in C_i} \max_{x \in C_i} \mathrm{dist}(x, c)$ while the cost of the clustering $\mathscr{C}$ is $\mathrm{cost}(\mathscr{C}) = \max_{i=1,\dots,k} \mathrm{cost}(C_i)$ and is to be minimized.

In the *k-diameter* problem we also have to find a partition of P into k clusters $\mathscr{C} = \{C_1, \dots, C_k\}$ and minimize the overall cost. However, we replace the cost of a cluster C_i by $\mathrm{cost}(C_i) = \max_{x,y \in C_i} \mathrm{dist}(x, y)$. For both the k-center and the k-diameter problem we denote by $\mathscr{O}_k$ an arbitrary but fixed *optimal clustering*.

We study the hierarchical version of the above problems, where we ask for a k-clustering $\mathscr{C}_k$ of P for every $1 \leq k \leq n$. The clusterings must be *hierarchically compatible*, which means that $\mathscr{C}_{k-1}$ is obtained from $\mathscr{C}_k$ by merging two of its clusters, i.e., for all $2 \leq k \leq n$ there are $A, B \in \mathscr{C}_k$ such that $\mathscr{C}_{k-1} = \mathscr{C}_k \backslash \{A, B\} \cup \{A \cup B\}$. A sequence of such k-clusterings $(\mathscr{C}_k)_{k=1}^n$ is called a *hierarchical clustering*. We say that it is an *α-approximation* if $\mathrm{cost}(\mathscr{C}_k) \leq \alpha \, \mathrm{cost}(\mathscr{O}_k)$ for all $1 \leq k \leq n$. Thus the task is to find a hierarchical clustering which is a good approximation to the optimal solution on every level k.

A common class of approaches for computing such hierarchical clusterings are *agglomerative linkage algorithms*. As outlined above, a hierarchical clustering can be computed in a bottom-up fashion, where pairs of clusters are merged successively. Agglomerative linkage procedures do exactly that, with the choice of clusters to be merged at every step given by a linkage function. Such a linkage function maps all possible pairs of disjoint clusters onto $\mathbb{R}_+$ and the algorithm chooses one pair that minimizes this value: Suppose that we have already constructed $\mathscr{C}_k$ and are using the linkage function f. Then $\mathscr{C}_{k-1}$ is given by merging a pair $A \neq B \in \mathscr{C}_k$ with $f(A, B) = \min_{A' \neq B' \in \mathscr{C}_k} f(A', B')$. As already stated, the two linkage functions we are interested in are:

- Single linkage: $(A, B) \mapsto \mathrm{dist}(A, B) = \min_{(a,b) \in A \times B} \mathrm{dist}(a, b)$.
- Complete linkage: $(A, B) \mapsto \mathrm{cost}(A \cup B)$.

To analyze the performance of the respective agglomerative algorithms we often consider the smallest clustering from $(\mathscr{C}_k)_{k=1}^n$ (in terms of the number of clusters) whose cost does not exceed a given bound. This perspective is already used by Großwendt and Röglin [8] and allows a better handling of the cost. For any $x \geq 0$ let $t_{\leq x} = \min\{k \mid \mathrm{cost}(\mathscr{C}_k) \leq x\}$ and set $\mathscr{H}_x = \mathscr{C}_{t_{\leq x}}$. Observe that $\mathscr{H}_x$ is the smallest clustering from $(\mathscr{C}_k)_{k=1}^n$ with cost at most x. Thus it has the useful property that every merge of two clusters in $\mathscr{H}_x$ results in a clustering of cost more than x. Furthermore, for a cluster $C \subseteq P$ and an optimal k-clustering $\mathscr{O} = \mathscr{O}_k$ we denote by $\mathscr{O}_C = \{O \in \mathscr{O} \mid O \cap C \neq \emptyset\}$ the set of all optimal k-clusters hit by C.

3 Approximation Guarantee of Single Linkage

As outlined in [6] there are clustering instances where single linkage builds chains yielding the lower bound $\Omega(k)$ on the approximation factor. We show in Appendix A that this is the worst case scenario, as in fact single linkage computes an $O(k)$-approximation for hierarchical k-center/k-diameter.

▶ **Theorem 1.** *Let $(\mathscr{C}_k)_{k=1}^n$ be the hierarchical clustering computed by single linkage on (P, dist) and let $\mathscr{O}_k$ be an optimal clustering for k-center or k-diameter, respectively. We have for all $1 \leq k \leq n$*

1. $\mathrm{cost}(\mathscr{C}_k) \leq (2k + 2) \cdot \mathrm{cost}(\mathscr{O}_k)$ *for the k-center cost*
2. $\mathrm{cost}(\mathscr{C}_k) \leq 2k \cdot \mathrm{cost}(\mathscr{O}_k)$ *for the k-diameter cost.*

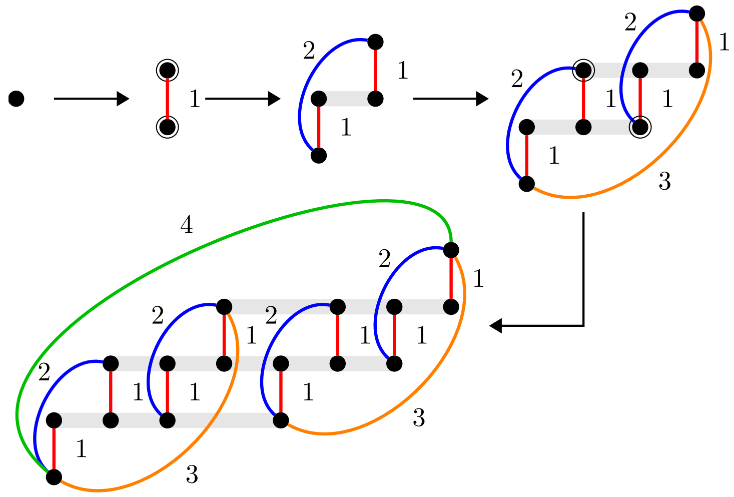

Figure 2 The progression of the first 5 components $K_1, \dots, K_5$. The gray sets indicate points on the same level and form the optimal clusters. When analyzing the instance for the radius, the encircled points in the K_2 and K_4 component indicate their optimal centers.

4 Lower Bounds for Complete Linkage

In the following we show that complete linkage performs asymptotically as bad as single linkage in the worst case. That is, for every $k \in \mathbb{N}$ we provide an instance P_k 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. Recall from the introduction 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 2.

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

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

4.1 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.** *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 $(V(P_k), \mathrm{dist})$ we slightly abuse our notation and see G_ℓ as a cluster with $\mathrm{cost}(G_\ell) = \max_{x,y \in V(G_l)} \mathrm{dist}(x, y)$, 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 4.** *Let G_ℓ be an ℓ-graph contained in P_k. We have* $\mathrm{cost}(G_\ell) = \ell - 1$.

Proof. We prove the upper bound $\mathrm{cost}(G_\ell) \leq \ell - 1$ 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 $\mathrm{dist}(s, t) = \mathrm{cost}(G_\ell)$. 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

$$\mathrm{cost}(G_\ell) = \mathrm{dist}(s, t) \leq \mathrm{cost}(G_{\ell-1}^{(0)}) \leq \ell - 2 < \ell - 1$$

by induction and we are done. Otherwise we may assume that $s \in V(G_{\ell-1}^{(0)})$ and $t \in V(G_{\ell-1}^{(1)})$. 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 $u \in V(G_{\ell-1}^{(1)})$. By induction $\mathrm{dist}(u, t) \leq \ell - 2$ and so

$$\mathrm{cost}(G_\ell) = \mathrm{dist}(s, t) \leq \mathrm{dist}(s, u) + \mathrm{dist}(u, t) \leq 1 + \ell - 2 = \ell - 1.$$

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

To see the lower bound $\mathrm{cost}(G_\ell) \geq \ell - 1$, we apply Lemma 3 to the unique point s with level 1 and the unique point t with level ℓ in G_ℓ. This shows that $\mathrm{cost}(G_\ell) \geq \mathrm{dist}(s, t) \geq \ell - 1$. ◀

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

▶ **Lemma 5.** *Complete linkage might merge clusters on $(V(P_k), \mathrm{dist})$ in such a way that for all $\ell \leq k$, the clustering $\mathscr{H}_{\ell-1}$ 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 $\mathscr{H}_{\ell-1}$ consists exactly of the ℓ-graphs of the instance. Since we are dealing with integer weights, any new merge increases the cost 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 cost from $\ell - 1$ to ℓ (see Lemma 4). 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. 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 $\mathrm{dist}(s,t) \geq \ell + 1$. In the second case we use Lemma 3 and obtain

$$\begin{aligned}
\mathrm{dist}(s,t) &= \mathrm{dist}(s,u) + \mathrm{dist}(u,w) + \mathrm{dist}(w,t) \\
&\geq |\phi_k(s) - \phi_k(u)| + 1 + |\phi_k(w) - \phi_k(t)| \\
&= |\phi_k(s) - \phi_k(t)| + 1 \\
&= \ell + 1.
\end{aligned}$$

It follows that $\mathscr{H}_\ell$ consists exactly of the $(\ell+1)$-graphs that make up P_k. ◀

Proof of Theorem 2 (diameter). Lemma 5 shows that $\mathscr{H}_{k-1}$ 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 $\mathscr{H}_{k-1}$, this last merge increases the cost 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. ◀

4.2 A Lower Bound for Radius-Based Costs

We show that the instance $(V(P_k), \mathrm{dist})$ 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 $\mathrm{cost}(G_\ell) = \min_{c \in V(G_\ell)} \max_{x \in V(G_\ell)} \mathrm{dist}(c,x)$, the radius of $V(G_\ell)$.

To prove Lemma 6 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 $2\ell - 1$.

▶ **Lemma 6.** *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 $\text{cost}(G_{2\ell}) = \ell$ and furthermore, all optimal centers that induce this cost are themselves already contained in $G_{2\ell}$ (and not in any other 2ℓ-graph).*

Proof. By Lemma 4 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 4 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

$$\text{dist}(c, x) \leq \text{dist}(c, y) + \text{dist}(y, x) \leq 1 + (\ell - 1) = \ell.$$

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 $\max_{x' \in V(G_{2\ell})} \text{dist}(c, x') \geq \ell + 1$. 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 $\text{dist}(c, x) \geq \ell + 1$. 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

$$\text{dist}(c, x) \geq 2\ell \geq \ell + 1$$

or it weights 1 and $\phi_k(u) = \phi_k(w)$, so

$$\begin{aligned} \text{dist}(c, x) &= \text{dist}(c, u) + \text{dist}(u, v) + \text{dist}(v, x) \\ &\geq |\phi_k(c) - \phi_k(u)| + 1 + |\phi_k(w) - \phi_k(x)| \\ &= |\phi_k(c) - \phi_k(x)| + 1 \\ &\geq \ell + 1. \end{aligned}$$

In case $\phi_k(c) \geq \lambda + \ell$ we can prove analogously that $\text{dist}(c, y) \geq \ell + 1$ 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 cost 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 7 ensures that the cost indeed increases.

▶ **Lemma 7.** *Let C, D be two subsets of $V(P_k)$ with $\text{cost}(C) = \text{cost}(D)$. 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 $\text{cost}(C \cup D) > \text{cost}(C)$.*

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

$$\max_{y \in C \cup D} \text{dist}(y, x) \geq \max_{y \in D} \text{dist}(y, x) > \text{cost}(D) = \text{cost}(C)$$

So we have for all $x \in V(P_k)$ that $\max_{y \in C \cup D} \text{dist}(y, x) > \text{cost}(C)$ which proves the lemma. ◀

Now, with this we can prove that the merging behavior of complete linkage reconstructs our components. Observe that Theorem 2 is an immediate consequence of Corollary 8.

▶ **Corollary 8.** *Complete linkage might merge clusters in* $(V(P_k), \mathrm{dist})$ *in such a way, that for* $1 \leq \ell \leq \frac{k}{2}$*, the clustering* $\mathscr{H}_\ell$ *consists exactly of the* 2ℓ*-graphs that make up* P_k*.*

Proof. The proof is an analogous induction to Lemma 5. Consider the case $\ell = 1$. The first merge increases the cost to 1. Observe by Lemma 6 that the cost 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 7 any further merge necessarily has to increase the cost. Hence $\mathscr{H}_1$ consists exactly of the 2-graphs.

Assume now that the claim holds for $\mathscr{H}_\ell$. The induction step works essentially the same as the base case. Any merge will increase the cost of the solution by at least 1 by definition of $\mathscr{H}_\ell$ 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 6). Furthermore, any additional merge would increase the cost to at least $\ell + 2$ (again by Lemma 7) and so $\mathscr{H}_{\ell+1}$ consist of the $(2\ell + 2)$-graphs. ◀

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 P_k we can show a lower bound of $\Omega(k)$ for both, diameter and radius, for any behavior of complete linkage on ties. For more details we refer to Appendix B.

5 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 authors' knowledge. We give an upper bound for complete linkage for hierarchical k-center and hierarchical k-diameter.

5.1 An Upper Bound for Radius-Based Cost

We show that the approximation ratio of the radius of any k-clustering $\mathscr{C}_k$ produced by complete linkage relative to an optimal k-center clustering is in $O(k)$.

▶ **Theorem 9.** *Let* $(\mathscr{C}_k)_{k=1}^n$ *be the hierarchical clustering computed by complete linkage on* (P, dist) *optimizing the radius. For all* $1 \leq k \leq n$ *the radius* $\mathrm{cost}(\mathscr{C}_k)$ *is upper bounded by* $O(k)\,\mathrm{cost}(\mathscr{O}_k)$*, where* $\mathscr{O}_k$ *is an optimal* k*-center clustering.*

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

We split the proof of Theorem 9 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 the work of Ackermann et al. [1], who use the same bound to estimate the cost of some few merge steps. Proposition 12 shows that the difference in cost between $\mathscr{C}_k$ and $\mathscr{C}_t$ for $t > k$ is at most $\lceil \log(t - k) \rceil + 1$. That is, $\mathrm{cost}(\mathscr{C}_k) \leq \lceil \log(t - k) \rceil + 1 + \mathrm{cost}(\mathscr{C}_t)$ holds for all $1 \leq k < t \leq n$. A clustering $\mathscr{C}_t$ whose cost we can estimate directly (i.e. without refering to any other clustering) thus yields a proper upper bound for $\mathrm{cost}(\mathscr{C}_k)$. Ideally, this clustering should consist of relatively few clusters (so that $\lceil \log(t - k) \rceil$ is small), while at the same time not being too expensive. Of course, however, these criteria oppose each other. Naively choosing the initial clustering $\mathscr{C}_t = \mathscr{C}_n$ is not good enough. Although its cost is minimal, the number of clusters is too high, only yielding an upper bound of $\mathrm{cost}(\mathscr{C}_k) \leq \lceil \log(n - k) \rceil + 1$. 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 cost by at most $2\,\mathrm{cost}(\mathscr{O}) = 1$ (Figure 1). 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 $\mathscr{H}_x$ are merged in $\mathscr{H}_{x+1}$.

▶ **Lemma 10.** *For all $x \geq 0$ the clustering $\mathscr{H}_{x+1}$ contains at most k clusters of cost at most x. In particular, it holds that $|\mathscr{H}_{x+1} \cap \mathscr{H}_x| \leq k$.*

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

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

because $\mathrm{dist}(d_1, d_2) \leq 2\,\mathrm{cost}(O) \leq 2\,\mathrm{cost}(\mathscr{O}) = 1$ and $\mathrm{cost}(D_i) \leq x$ for $i = 1, 2$. This contradicts the definition of $\mathscr{H}_{x+1}$, as D_1 and D_2 can still be merged without pushing the cost beyond $x+1$. ◀

With this we can upper bound $|\mathscr{H}_{x+i}|$ in terms of $|\mathscr{H}_x|$ for all $i \in \mathbb{N}$. The proof of Corollary 11 can be found in Appendix C.

▶ **Corollary 11.** *For all $i \in \mathbb{N}_+$ and $x \geq 0$ it holds that $|\mathscr{H}_{x+i}| \leq k + \frac{1}{2^i}(|\mathscr{H}_x| - k)$.*

▶ **Proposition 12.** *For all $k < t \leq n$ it holds that $\mathrm{cost}(\mathscr{C}_k) \leq \lceil \log(t-k) \rceil + 1 + \mathrm{cost}(\mathscr{C}_t)$.*

Proof. Let $x = \mathrm{cost}(\mathscr{C}_t)$, so that $\mathscr{H}_x$ consists of at most t clusters. Applying Corollary 11 with $i = \lceil \log(t-k) \rceil + 1$ then shows that

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

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

Part 2: A cheap clustering with few clusters

Suppose that there exists a complete linkage clustering $\mathscr{C}_t$ for some $t > k$ with $t \in O(2^k)$ clusters and $\mathrm{cost}(\mathscr{C}_t) \in O(k)$. Then applying Proposition 12 shows that

$$\mathrm{cost}(\mathscr{C}_k) \in \log(O(2^k)) + 1 + O(k) = O(k) = O(k)\,\mathrm{cost}(\mathscr{O})$$

and Theorem 9 is proven (recall that $\mathrm{cost}(\mathscr{O}) = \frac{1}{2}$). We show that $\mathscr{C}_t = \mathscr{H}_{4k+2}$ is a sufficiently good choice. To estimate the size of $\mathscr{H}_{4k+2}$, we distinguish between active and inactive clusters. Remember that $\mathscr{O}_C = \{O \in \mathscr{O} \mid O \cap C \neq \emptyset\}$ is the set of optimal clusters hit by C.

▶ **Definition 13.** *We call a cluster $C \in \mathscr{H}_x$* active, *if $\mathrm{cost}(C) \leq 4 \cdot |\mathscr{O}_C|$, or if there exists an active cluster $C' \in \mathscr{H}_{x-1}$ such that $\mathscr{O}_C \subseteq \mathscr{O}_{C'}$. Otherwise, C is called* inactive.

The behavior that makes complete linkage more difficult to analyze than single linkage is that the former sometimes merges clusters that are quite far apart. That is, contrary to single linkage, 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 cost of the clustering reaches $4k+2$, this number can also be bounded as we see in the following lemma.

▶ **Lemma 14.** *There are at most 2^k active clusters in $\mathscr{H}_{4k+2}$.*

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

$$\operatorname{cost}(C_1 \cup C_2) \leq \operatorname{cost}(C_2) + 1 \leq 4|\mathscr{O}_{C_2}| + 1 \leq 4k + 1$$

and so C_1 and C_2 would have been merged in $\mathscr{H}_{4k+2}$. Now, if there are more than 2^k active clusters in $\mathscr{H}_{4k+2}$, 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 15.** *There are at most $4k^2 + k$ inactive clusters in $\mathscr{H}_{4k+2}$.*

Proof. Let m_x be the number of inactive clusters in $\mathscr{H}_x$. 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 $D \in \mathscr{H}_x$ be an inactive cluster. Let $D_1, \dots, D_\ell \in \mathscr{H}_{x-1}$ 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 cost 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 $4 \cdot |\mathscr{O}_{D_i}| \geq \operatorname{cost}(D_i) \geq x - 2$. Right away, notice that $|\mathscr{O}_{D_i}| < |\mathscr{O}_D|$ since otherwise D would also be active by definition. But then

$$\operatorname{cost}(D) \leq x \leq \operatorname{cost}(D_i) + 2 \leq 4|\mathscr{O}_{D_i}| + 2 < 4(|\mathscr{O}_{D_i}| + 1) \leq 4|\mathscr{O}_D|$$

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

$$\{D \in \mathscr{H}_{x-1} \mid D \text{ is inactive}\} \cup \{D \in \mathscr{H}_{x-1} \mid \operatorname{cost}(D) < x - 2\}.$$

The cardinality of the set on the left is m_{x-1} and, by Lemma 10, the cardinality of the set on the right is at most k. This proves the claim. ◀

▶ **Corollary 16.** *$\mathscr{H}_{4k+2}$ consists of at most $2^k + 4k^2 + k$ clusters.*

Notice that Theorem 9 is an immediate consequence of Corollary 16 and Proposition 12.

5.2 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 k-center objective by the k-diameter objective is to deal with the possibly large increase of cost after a merge step (see Figure 1).

For some arbitrary but fixed k let $\mathscr{O}$ denote an optimal k-diameter solution and assume that $\operatorname{cost}(\mathscr{O}) = 1$ from now on. To motivate our approach consider the clustering $\mathscr{H}_1$ computed by complete linkage at time $t_{\leq 1}$. Observe that every optimal cluster can fully contain at most one cluster from $\mathscr{H}_1$, as the union of such clusters would cost at most 1. Now, consider the graph $G = (V, E)$ with $V = \mathscr{O}$ and edges $\{A, B\} \subset V$ for every cluster $C \in \mathscr{H}_1$ 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 $\mathscr{H}_1 = \mathscr{O}$ 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 $|V(Z)|^2$. This leads to an upper bound of k^2 for all clusters C constructed by complete linkage with $C \subset \cup_{A \in V(Z)} A$. We show that the number of clusters which do not admit this property is sufficiently small, such that in the end, we are able to prove that $\mathscr{H}_{k^2}$ consists of at most k clusters. This immediately leads the following theorem.

▶ **Theorem 17.** *Let $(\mathscr{C}_k)_{k=1}^n$ be the hierarchical clustering computed by complete linkage on (P, dist) optimizing the diameter. For all $1 \leq k \leq n$ the diameter $\operatorname{cost}(\mathscr{C}_k)$ is upper bounded by $k^2 \operatorname{cost}(\mathscr{O}_k)$, where $\mathscr{O}_k$ is an optimal k-diameter clustering.*

Essential for this section is a sequence of *cluster graphs* $G_t = (V_t, E_t)$ for $t = 1, \ldots, k^2$ constructed directly on the set $V_t = \mathscr{O}$ of optimal k-clusters. We start with the cluster graph G_1 that contains edges $\{A, B\}$ for every two vertices $A, B \in V_1 = \mathscr{O}$ that are hit by a common cluster from $\mathscr{H}_1$. To this we successively add edges based on a vertex labeling in order to create the remaining cluster graphs $G_2, \ldots, G_{k^2}$. 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 18. 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 $\mathscr{H}_{t+1}$, we add an edge $\{A, B\}$ to E_{t+1}.

▶ **Definition 18.** *Let $A \in V_{t+1}$ 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 $\lceil \operatorname{cost}(Z_A) \rceil \leq t$, and* active *otherwise. Here, and in the following $\operatorname{cost}(Z_A) = \operatorname{cost}(\bigcup_{B \in V(Z_A)} B)$ denotes the cost of merging all optimal clusters contained in $V(Z_A)$.*

Thus if a connected component in G_t has small cost, then all vertices in this component become inactive in G_{t+1} by definition. We state the following useful properties of inactive vertices in $(G_t)_{t=1}^{k^2}$.

▶ **Lemma 19.** *If Z is a connected component in G_{t+1} with $V(Z) \cap V^i_{t+1} \neq \emptyset$, then*

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

Moreover we have $V^i_t \subseteq V^i_{t+1}$, so once vertices become inactive, they stay inactive. Equivalently, $V^a_{t+1} \subseteq V^a_t$.

Proof. Take any inactive vertex $A \in V^i_{t+1} \cap V(Z)$ and consider the connected component Z_A in G_t containing A. By Definition 18, we have that $\lceil \mathrm{cost}(Z_A) \rceil \leq t$ and so all other vertices in Z_A have to be in V^i_{t+1} as well. We observe that $E_{t+1} \setminus E_t$ only contains edges between vertices from V^a_{t+1} by construction. This shows $Z = Z_A$.

It is left to show that inactive vertices stay inactive. For $t = 1$ the inactive vertices V^i_1 are already connected components with cost at most 1. As such, they remain inactive at step $t = 2$. For $t \geq 2$, consider an inactive vertex $A \in V^i_t$ and the connected component $Z \subseteq G_t$ containing it. We showed previously that $V(Z) \subset V^i_t$ and so Z is also a connected component in G_{t+1} with $\lceil \mathrm{cost}(Z) \rceil \leq t - 1 < t$ and thus $A \in V(Z) \subset V^i_{t+1}$. ◀

▶ **Definition 20.** *Let $C \in \mathscr{H}_t$ for some fixed $t \in \mathbb{N}$. We define $\mathscr{I}_t = \{C \in \mathscr{H}_t \mid \mathscr{O}_C \cap V^i_t \neq \emptyset\}$ as the set of all clusters in $\mathscr{H}_t$ which hit at least one inactive vertex of G_t. We call these clusters inactive and all clusters from $\mathscr{H}_t \backslash \mathscr{I}_t$ active.*

We prove the following easy property about active clusters.

▶ **Lemma 21.** *If $C \in \mathscr{H}_t \setminus \mathscr{I}_t$, then $G_t[\mathscr{O}_C]$ forms a clique. In particular there exists a connected component in G_t that fully contains $\mathscr{O}_C$.*

Proof. By definition of $\mathscr{I}_t$, $\mathscr{O}_C$ must consist exclusively of active vertices. Since all of them are hit by $C \in \mathscr{H}_t$ there exists an edge $\{A, B\} \in E_t$ for every pair $A, B \in \mathscr{O}_C$. In other words, $G_t[\mathscr{O}_C]$ 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 $|V^i_t|$.

▶ **Lemma 22.** *The number of inactive clusters in $\mathscr{H}_t$ is at most the number of inactive vertices at time t. That is, $|\mathscr{I}_t| \leq |V^i_t|$ 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 $\phi_t : \mathscr{I}_t \to V^i_t$:

- Let $C \in \mathscr{I}_1$ be an inactive cluster. By definition C thus has to intersect an inactive optimal cluster $A \in V^i_1$. 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 $\mathscr{O}_C = \{\phi_1(C)\}$.
- 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 $A \in \mathscr{O}_C \cap V^i_t$ 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 $\phi_{t-1}(D) \in V^i_{t-1} \subset V^i_t$ by Lemma 19, this shows that ϕ_t really maps into V^i_t.

Suppose that there exist two inactive clusters $C, D \in \mathscr{I}_1$ that are mapped to the same inactive vertex $A \in V^i_1$. Then, by the construction of ϕ_1, $\mathscr{O}_C = \{A\} = \mathscr{O}_D$ 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 $\mathscr{H}_1$. 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 $\mathscr{H}_{t-1} \setminus \mathscr{I}_{t-1}$ and so there exist clusters $C', D' \in \mathscr{H}_{t-1} \setminus \mathscr{I}_{t-1}$ contained in C and D, respectively, such that $A = \phi_t(C) = \phi_t(D) \in \mathscr{O}_{C'} \cap \mathscr{O}_{D'}$. Applying Lemma 21 yields the existence of a connected component Z in G_{t-1} with $V(Z) \supset \mathscr{O}_{C'} \cup \mathscr{O}_{D'}$. We show that this connected component has cost at most $t-1$. In that case, C' and D' should have already been merged in $\mathscr{H}_{t-1}$; a contradiction. To show that $\operatorname{cost}(Z) \leq t-1$, consider the connected component Z' in G_t containing $A = \phi_t(C) = \phi_t(D) \in \mathscr{O}_C \cap \mathscr{O}_D \cap V_t^i$. Since A was chosen from a subset of V_t^i, we know from Lemma 19 that Z' is also a connected component in G_{t-1} with $\operatorname{cost}(Z') \leq t-1$. 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 $\mathscr{H}_{t-1} \setminus \mathscr{I}_{t-1}$, we know that $\mathscr{O}_C \subseteq V_{t-1}^a$. 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 $\mathscr{H}_t$ are nicely represented by the graph G_t as it is shown in Lemma 21. We can indirectly bound the cost of active clusters by bounding the cost of the connected components they are contained in.

▶ **Lemma 23.** *Let Z be a connected component in G_t. If $V(Z) \subset V_t^a$, we have* $\operatorname{cost}(Z) \leq |V(Z)|^2$.

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 $A = Q_1, \dots, Q_s = B$ 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 $\operatorname{dist}(p_j, q_j) \leq 1$. Using the triangle inequality we obtain

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

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

$$\operatorname{cost}(Z) \leq 2|V(Z)| - 1 \leq |V(Z)|^2.$$

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 $j, j' \in \{1, \dots, u\}$. We observe that $V(Z_j) \subset V(Z) \subset V_t^a \subset V_{t-1}^a$. Thus we obtain by induction that

$$\operatorname{cost}(Z_j) \leq |V(Z_j)|^2. \tag{1}$$

Suppose that $\lceil \operatorname{cost}(Z_j) \rceil \leq t-1$. 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

$$t \leq \lceil \operatorname{cost}(Z_j) \rceil. \tag{2}$$

Combining (1) and (2) we obtain

$$t \leq \sqrt{\lceil \operatorname{cost}(Z_j) \rceil \lceil \operatorname{cost}(Z_{j'}) \rceil} \leq \sqrt{|V(Z_j)|^2 |V(Z_{j'})|^2} = |V(Z_j)||V(Z_{j'})|. \tag{3}$$

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 $1 = m_1 < m_2 < \ldots < m_l = s$ 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 $j \in \{1, \dots, l\}$. Since $(Q_{m_j-1}, Q_{m_j}) \in E_t$ we know that there exists a cluster in $\mathscr{H}_t$ 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 $\operatorname{dist}(p_j, q_j) \leq t$. We obtain

$$\begin{aligned}\operatorname{dist}(p,q) &\leq \sum_{j=1}^{l-1} \big(\operatorname{cost}(Z^{(j)}) + \operatorname{dist}(p_j, q_j) \big) + \operatorname{cost}(Z^{(l)}) \leq \sum_{j=1}^{l} \big(|V(Z^{(j)})|^2 + t \big) \\ &\leq \Big(\sum_{j=1}^{l} |V(Z^{(j)})| \Big)^2 = |V(Z)|^2.\end{aligned}$$

For the second inequality we use (1) and $\operatorname{dist}(p_j, q_j) \leq t$. For the third inequality we use (3). So we obtain the claimed upper bound on the cost of Z. ◀

We see that a connected component in G_{k^2} cannot contain two active clusters, yielding the following upper bound.

▶ **Lemma 24.** *At time $t_{\leq k^2}$ the number of active clusters is less than or equal to the number of active vertices. In other words, $|\mathscr{H}_{k^2} \setminus \mathscr{I}_{k^2}| \leq |V_{k^2}^a|$.*

Proof. By Lemma 21 we know that every cluster $C \in \mathscr{H}_{k^2} \setminus \mathscr{I}_{k^2}$ is fully contained in a connected component Z_C from G_{k^2}. We show that mapping any such C to an arbitrary vertex in Z_C yields an injective map $\varphi : \mathscr{H}_{k^2} \setminus \mathscr{I}_{k^2} \hookrightarrow V_{k^2}^a$. First, notice that φ is well-defined: If Z_C contains an inactive vertex, then all its vertices are inactive (Lemma 19), contradicting the choice of C as active.

Suppose now that there are two different clusters $C, C' \in \mathscr{H}_{k^2} \setminus \mathscr{I}_{k^2}$ 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 23), that $\operatorname{cost}(Z_C) \leq |V(Z_C)|^2 \leq k^2$ and so C and C' would have already been merged in $\mathscr{H}_{k^2}$. 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.

Proof of Theorem 17. Using Lemma 22&24 we obtain $|\mathscr{H}_{k^2}| = |\mathscr{H}_{k^2} \setminus \mathscr{I}_{k^2}| + |\mathscr{I}_{k^2}| \leq |V_{k^2}^a| + |V_{k^2}^i| = k$, yielding $\operatorname{cost}(\mathscr{C}_k) \leq \operatorname{cost}(\mathscr{H}_{k^2}) \leq k^2 \operatorname{cost}(\mathscr{O}_k)$. ◀

References

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 Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. *SIAM J. Comput.*, 49(4), 2020. doi:10.1137/18M1171321.

3 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.

4 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.

5 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.

6 Sanjoy Dasgupta and Philip M. Long. Performance guarantees for hierarchical clustering. *J. Comput. Syst. Sci.*, 70(4):555–569, 2005. doi:10.1016/j.jcss.2004.10.006.

7 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.

8 Anna Großwendt and Heiko Röglin. Improved analysis of complete-linkage clustering. *Algorithmica*, 78(4):1131–1150, 2017. doi:10.1007/s00453-017-0284-6.

9 Anna Großwendt, Heiko Röglin, and Melanie Schmidt. Analysis of ward's method. In Timothy M. Chan, editor, *Proc. of the Thirtieth Annu. ACM-SIAM Symp. on Discrete Algorithms, SODA*, pages 2939–2957. SIAM, 2019. doi:10.1137/1.9781611975482.182.

10 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.

11 D. Ellis Hershkowitz and Gregory Kehne. Reverse greedy is bad for k-center. *Inf. Process. Lett.*, 158:105941, June 2020. doi:10.1016/j.ipl.2020.105941.

12 Dorit S. Hochbaum. When are np-hard location problems easy? *Ann. Oper. Res.*, 1(3):201–214, 1984. doi:10.1007/BF01874389.

13 Dorit S. Hochbaum and David B. Shmoys. A best possible heuristic for the k-center problem. *Math. Oper. Res.*, 10(2):180–184, 1985. doi:10.1287/moor.10.2.180.

14 Wen-Lian Hsu and George L. Nemhauser. Easy and hard bottleneck location problems. *Discret. Appl. Math.*, 1(3):209–215, 1979. doi:10.1016/0166-218X(79)90044-1.

15 Guolong Lin, Chandrashekhar Nagarajan, Rajmohan Rajaraman, and David P. Williamson. A general approach for incremental approximation and hierarchical clustering. *SIAM J. Comput.*, 39(8):3633–3669, 2010. doi:10.1137/070698257.

16 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.

A Single Linkage

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

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

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

To estimate the cost of a single linkage cluster $C \in \mathscr{C}_k$ 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 $2\operatorname{cost}(\mathscr{O})$ or $\operatorname{cost}(\mathscr{O}')$ when considering the radius or diameter, respectively. One can already anticipate that this gives an upper bound of $O(k)$ on the radius or diameter of any such cluster C. In Figure 3 we see an example of such a cluster C and the optimal clusters hit by C.

▶ **Lemma 25.** *Let $C \in \mathscr{C}_t$ be a cluster computed by single linkage at a time step $t \geq k$. Then the graphs $G[\mathscr{O}_C]$ and $G'[\mathscr{O}'_C]$ induced by the vertex set of optimal clusters hit by C are connected.*

Proof. We prove the lemma for $G[\mathscr{O}_C]$ by induction. At the beginning ($t = n$) the lemma obviously holds, since any cluster contained in $\mathscr{C}_n$ is a point and thus hits only one single optimal cluster. Assume now that the claim holds for $t > k$. By the pidgeonhole principle there must exist two clusters $C, C' \in \mathscr{C}_t$ with two points $c \in C$ and $c' \in C'$ lying in the same optimal cluster $O \in \mathscr{O}$. We know that $\operatorname{dist}(C, C') \leq 2\operatorname{cost}(O) \leq 2\operatorname{cost}(\mathscr{O})$. 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 $G[\mathscr{O}_D]$ and $G[\mathscr{O}_{D'}]$ are connected finishes the proof. One proves analogously that $G'[\mathscr{O}'_C]$ is connected. ◀

As we see in Figure 3 this already yields an upper bound of $2k\operatorname{cost}(\mathscr{O}')$ on the diameter of C. We estimate the radius of C by looking at the paths going through optimal clusters in $\mathscr{O}_C$ that are at distance at most $2\operatorname{cost}(\mathscr{O})$ from one another. Choosing the center appropriately and uncoiling these paths in our original space P yields our upper bound of $(2k+2)\operatorname{cost}(\mathscr{O})$.

▶ **Theorem 1.** *Let $(\mathscr{C}_k)_{k=1}^n$ be the hierarchical clustering computed by single linkage on (P, dist) and let $\mathscr{O}_k$ be an optimal clustering for k-center or k-diameter, respectively. We have for all $1 \leq k \leq n$*

1. $\operatorname{cost}(\mathscr{C}_k) \leq (2k+2) \cdot \operatorname{cost}(\mathscr{O}_k)$ *for the k-center cost*
2. $\operatorname{cost}(\mathscr{C}_k) \leq 2k \cdot \operatorname{cost}(\mathscr{O}_k)$ *for the k-diameter cost.*

Proof. We prove the statement for k-center. Fix an arbitrary time step $1 \leq k \leq n$ and denote $\mathscr{O} = \mathscr{O}_k$. Let $C \in \mathscr{C}_k$ be an arbitrary cluster and P a longest simple path in $G[\mathscr{O}_C]$. Choose as center for C an arbitrary vertex $c \in C \cap O$ from an optimal cluster O lying in the middle of P. Note that by this choice every other vertex in $G[\mathscr{O}_C]$ is reachable from O by a path of length at most $\frac{k}{2}$. Uncoiling such paths in P gives us an upper bound of $2(k+1)\operatorname{cost}(\mathscr{O})$ for the distance between c and any other point $z \in C$ as follows: If $O_z \in \mathscr{O}$ 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 $G[\mathscr{O}_C]$ of length $\ell \leq \frac{k}{2}$ connecting them. That means, for each $i = 1, \ldots, \ell$ there exist points $x_i \in O_i, y_{i+1} \in O_{i+1}$ such that $\operatorname{dist}(x_i, y_{i+1}) \leq 2\operatorname{cost}(\mathscr{O})$. Hence

$$\begin{aligned}\operatorname{dist}(c, z) \leq \operatorname{dist}(c, x_1) &+ \sum_{i=1}^{\ell-1} (\operatorname{dist}(x_i, y_{i+1}) + \operatorname{dist}(y_{i+1}, x_{i+1})) \\ &+ \operatorname{dist}(x_\ell, y_{\ell+1}) + \operatorname{dist}(y_{\ell+1}, z) \\ &\leq 2(2\ell+1)\operatorname{cost}(\mathscr{O}) \leq 2(k+1)\operatorname{cost}(\mathscr{O}).\end{aligned}$$

Using Lemma 25 one proves the statement for k-diameter analogously. ◀

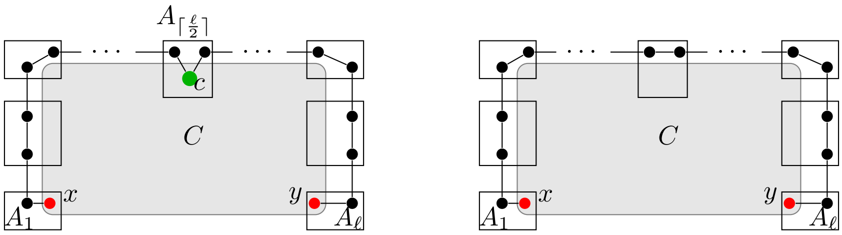

Figure 3 Cluster C hits the optimal clusters $A_1, \ldots, A_\ell$ with $\operatorname{dist}(A_i, A_{i+1}) \leq 2\operatorname{cost}(\mathscr{O})$ when considering the radius on the left and $\operatorname{dist}(A_i, A_{i+1}) \leq \operatorname{cost}(\mathscr{O}')$ when considering the diameter on the right. In the left picture, we see that choosing center c in $A_{\lceil \frac{\ell}{2} \rceil}$ leads to radius $\leq 2(\ell+1)\operatorname{cost}(\mathscr{O})$. Similarly the diameter of C in the right picture is at most $2\ell \operatorname{cost}(\mathscr{O})$.

B A Lower Bound for Complete Linkage without Bad Ties

In this section we focus on modifying the instance $(V(P_k), \operatorname{dist})$ such that merging two ℓ-graphs G_ℓ, G'_ℓ which are part of the same $(\ell+1)$-graph is slightly cheaper than performing any other merge in a clustering consisting of all ℓ-graphs.

B.1 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 point $s \in V(G_{k-1}^{(0)})$ 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 P_k 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 $(V(P_k), \operatorname{dist})$ at time $t_{\leq \ell(1-\epsilon)}$ consists exactly of the $(\ell+1)$-graphs that make up the instance.

▶ **Lemma 26.** *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 27.** *Let G_ℓ be an ℓ-graph contained in P_k. We have $\operatorname{cost}(G_\ell) = (\ell-1)(1-\epsilon)$.*

Proof. We prove the upper bound $\operatorname{cost}(G_\ell) \leq (\ell-1)(1-\epsilon)$ 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 $\operatorname{cost}(G_\ell) = \operatorname{dist}(s, t)$. 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

$$\operatorname{dist}(s,t) \le \operatorname{cost}(G^{(0)}_{\ell-1}) \le (\ell-2)(1-\epsilon) < (\ell-1)(1-\epsilon)$$

by induction and we are done. Otherwise we may assume that $s \in V(G^{(0)}_{\ell-1})$ and $t \in V(G^{(1)}_{\ell-1})$. 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 $\ell \le 2$. Using $\epsilon \le \frac{1}{2}$ and $\ell \ge 3$ we obtain

$$\operatorname{dist}(s,t) = 1 \le (\ell-1)(1-\epsilon).$$

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

$$\operatorname{cost}(G_\ell) = \operatorname{dist}(s,t) \le (\ell-1)(1-\epsilon).$$

To see the lower bound $\operatorname{cost}(G_\ell) \ge (\ell-1)(1-\epsilon)$, we apply Lemma 26 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 $\operatorname{cost}(G_\ell) \ge \operatorname{dist}(s,t) \ge (\ell-1)(1-\epsilon)$. ◀

We show that complete linkage must reconstruct these components as clusters.

► **Lemma 28.** *Complete linkage must merge clusters on $(V(P_k), \operatorname{dist})$ in such a way that for all $\ell < k$, the clustering $\mathscr{H}_{\ell(1-\epsilon)}$ 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 $\mathscr{H}_{(\ell-1)(1-\epsilon)}$ 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 27, that cost $\operatorname{cost}(G_{\ell+1}) = \ell(1-\epsilon)$.

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 $\ge (\ell+1)(1-\epsilon)$.
2. There exist $s \in V(G_\ell)$ and $t \in V(G'_\ell)$ with $|\phi_k(s) - \phi_k(t)| \ge \ell - 1$.

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 $\operatorname{dist}(s,t) > \ell(1-\epsilon)$ 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 $\ge (\ell+1)(1-\epsilon)$ or u and w are on the same level and the edge is of weight 1. In the first case we conclude

$$\operatorname{dist}(s,t) \ge (\ell+1)(1-\epsilon) > \ell(1-\epsilon).$$

In the second case we obtain that

$$\begin{aligned}\operatorname{dist}(s,t) &= \operatorname{dist}(s,u) + 1 + \operatorname{dist}(w,t)\\ &\geq |\phi_k(s) - \phi_k(u)|(1-\epsilon) + 1 + |\phi_k(w) - \phi_k(t)|(1-\epsilon)\\ &= |\phi_k(s) - \phi_k(t)|(1-\epsilon) + 1\\ &\geq (\ell-1)(1-\epsilon) + 1\\ &> \ell(1-\epsilon).\end{aligned}$$

We see that $\mathscr{H}_{\ell(1-\epsilon)}$ must consists exactly of the $(\ell+1)$-graphs of P_k. ◀

Lemma 28 shows that $\mathscr{H}_{(k-1)(1-\epsilon)}$ 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 29.** *However the tie-breaks are resolved, complete linkage computes a k-clustering on $(V(P_k), \operatorname{dist})$ with diameter $(k-1)(1-\epsilon)$ while the optimal k-clustering has diameter 1.*

B.2 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 P_k 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 26 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 30.** *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 $\operatorname{cost}(G_{2\ell}) = \ell(1-\epsilon)$. 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 26 we know that $\max\{\operatorname{dist}(s,x), \operatorname{dist}(t,x)\} \geq \ell(1-\epsilon)$ and therefore $\operatorname{cost}(G_{2\ell}) \geq \ell(1-\epsilon)$.

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 $\operatorname{dist}(x,c) \leq \ell(1-\epsilon)$.

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 $\operatorname{dist}(x,y) \geq (\ell-1)(1-\epsilon)+1$. 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 26 and obtain

$$\begin{aligned}\operatorname{dist}(x, y) &= \operatorname{dist}(x, u) + \operatorname{dist}(u, w) + \operatorname{dist}(w, y)\\ &\geq |\phi_k(x) - \phi_k(u)|(1 - \epsilon) + 1 + |\phi_k(w) - \phi_k(y)|(1 - \epsilon)\\ &= |\phi_k(x) - \phi_k(y)|(1 - \epsilon) + 1\\ &\geq \ell(1 - \epsilon) + 1\end{aligned}$$

and in the second case we obtain

$$\operatorname{dist}(x, y) \geq (\ell + 2)(1 - \epsilon) \geq \ell(1 - \epsilon) + 1. \qquad ◀$$

This immediately leads the following results.

▶ **Corollary 31.** *Complete linkage must merge clusters on $(V(P_k), \operatorname{dist})$ in such a way that for all $1 \leq \ell \leq \frac{k}{2}$, the clustering $\mathscr{H}_{\ell(1-\epsilon)}$ consists exactly of the 2ℓ-graphs that make up P_k.*

▶ **Corollary 32.** *However the tie-breaks are resolved, complete linkage computes a k-clustering on $(V(P_k), \operatorname{dist})$ with radius $\frac{k}{2}(1 - \epsilon)$, while the optimal k-clustering has radius 1.*

C An Upper Bound for Radius-Based Cost

▶ **Corollary 11.** *For all $i \in \mathbb{N}_+$ and $x \geq 0$ it holds that $|\mathscr{H}_{x+i}| \leq k + \frac{1}{2^i}(|\mathscr{H}_x| - k)$.*

Proof. First, we consider what happens when we increase the cost by 1. We fix an arbitrary $x' \geq 0$. Lemma 10 shows that at most k clusters from $\mathscr{H}_{x'}$ are left untouched, while the remaining $|\mathscr{H}_{x'}| - k$ clusters have to be merged with at least one other cluster (thus at least halving the number of those clusters) to get to $\mathscr{H}_{x'+1}$. This yields a bound of

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

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

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

Substituting this into the inequality

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

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

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

as claimed. ◀