Polynomial Time Approximation Schemes for Clustering in Low Highway Dimension Graphs*

Andreas Emil Feldmann†
Charles University, Prague, Czechia
feldmann.a.e@gmail.com

David Saulpic,
LIP6, Sorbonne Université, Paris, France
david.saulpic@lip6.fr

Abstract

We study clustering problems such as k-Median, k-Means, and Facility Location in graphs of low highway dimension, which is a graph parameter modeling transportation networks. It was previously shown that approximation schemes for these problems exist, which either run in quasi-polynomial time (assuming constant highway dimension) [Feldmann et al. SICOMP 2018] or run in FPT time (parameterized by the number of clusters k, the highway dimension, and the approximation factor) [Becker et al. ESA 2018, Braverman et al. SODA 2021]. In this paper we show that a polynomial-time approximation scheme (PTAS) exists (assuming constant highway dimension). We also show that the considered problems are NP-hard on graphs of highway dimension 1.

*A preliminary version of this paper appeared at ESA 2020 [18].
†Supported by the Czech Science Foundation GAČR (grant #19-27871X), and by the Center for Foundations of Modern Computer Science (Charles Univ. project UNCE/SCI/004).

1 Introduction

Clustering is a standard optimization task that seeks a "good" partition of a metric space, such that two points that are "close" should be in the same part. A good clustering of a dataset allows to retrieve and exploit data, and is therefore a common routine in data analysis. The underlying data can come from various sources and represent many different objects. In particular, it is often interesting to cluster geographic data. In that case, the metric space can be given by a transportation network, which can be modeled by graphs with low highway dimension.

In this article, we study some popular clustering objectives, namely FACILITY LOCATION, k-MEDIAN, and k-MEANS, in graphs with constant highway dimension. The two latter problems seek to find a set S of k points called *centers* in a metric (V, dist) that minimizes the function $\sum_{v\in V}(\min_{f\in S}\text{dist}(v,f))^p$, with $p=1$ for k-MEDIAN and $p=2$ for k-MEANS. The objective for FACILITY LOCATION is slightly different: each point f of the metric space has an *opening cost* w_f, and the goal is to find a set S that minimizes $\sum_{f\in S} w_f + \sum_{v\in V}\min_{f\in S}\text{dist}(v,f)$. These problems are APX-hard in general metric spaces, see for example Guha and Khuller [20] for FACILITY LOCATION, Jain et al. [23] for k-MEDIAN, and Awasthi et al. [6] for k-MEANS.

To bypass the hardness of approximation known for these problems, researchers have considered low dimensional input, such as Euclidean spaces of fixed dimension, metrics with bounded doubling dimension, or metrics arising from classes of minor-free graphs. Many algorithmic tools were developed for that purpose: in their seminal work, Arora et al. [5] gave the first polynomial time approximation scheme (PTAS) for Euclidean k-MEDIAN in $\mathbb{R}^2$, which generalizes to a quasi-polynomial time approximation scheme (QPTAS) in $\mathbb{R}^d$ for fixed d. This result was generalized by Talwar [26], who gave a QPTAS for metrics with bounded doubling dimension, and more recently by Cohen-Addad et al. [12], who gave a near-linear time approximation scheme for this setting.

In this work we focus on transportation networks, for which it can be argued that metric spaces with bounded doubling dimension are not a suitable model: for instance, hub-and-spoke networks seen in air traffic networks do not have low doubling dimension. Therefore we study graphs with constant *highway dimension*, which formalize structural properties of such networks. The following definition is taken from Feldmann et al. [17]. Here the *ball* $\beta_v(r)$ of radius r around $v\in V$ is the set of all vertices at distance at most r from v.

Definition 1. *The* highway dimension *of a graph G is the smallest integer h such that, for some universal constant $c>4$, for every $r\in\mathbb{R}^+$ and $v\in V$ there are at most h vertices in the ball $\beta_v(cr)$ of radius cr around v hitting all shortest paths of length more than r that lie in $\beta_v(cr)$.*

Before this work, for this class of graphs the only known approximation algorithms for clustering that compute $(1+\varepsilon)$-approximations for any $\varepsilon>0$ either run in quasi-polynomial $n^{(\log n)^{O(\log^2(h/\varepsilon))}}$ time, i.e., QPTASs [17], or in $2^{\tilde{O}(h^{O(1/\varepsilon)}+k)}\cdot n^{O(1)}$ time, i.e., parameterized approximation schemes [8, 10]. Thus an open problem was to identify polynomial-time approximation schemes (PTASs) for clustering in graphs of constant highway dimension.

1.1 Our results

Our main result is a PTAS for clustering problems on graphs of constant highway dimension. For convenience, we define slightly more general problems than those stated above. The k-CLUSTERING^q problem is defined as follows. An instance $\mathcal{I}$ consists of a metric (V,dist), a set of *facilities* (or *centers*) $F\subseteq V$, and a *demand function* $\chi: V\to\mathbb{N}_0$. The goal is to find a set $S\subseteq F$ with $|S|\le k$ minimizing $\sum_{v\in V}\chi(v)\cdot\min_{f\in S}\text{dist}(v,f)^q$. We call all vertices $v\in V$ with $\chi(v)>0$ the *clients* of $\mathcal{I}$. k-MEDIAN and k-MEANS are special cases of k-CLUSTERING^q, where $q=1$ and $q=2$.

The input to the FACILITY LOCATION^q problem is the same as for k-CLUSTERING^q, but additionally each facility $f \in F$ has an *opening cost* $w_f \in \mathbb{R}^+$. The goal is to find a set $S \subseteq F$ minimizing $\sum_{f\in S} w_f + \sum_{v\in V} \chi(v) \cdot \min_{f\in S} \mathrm{dist}(v, f)^q$. FACILITY LOCATION is a special case of FACILITY LOCATION^q, where $q = 1$.

Our main theorem is the following, where $X = \max_{v\in V} \chi_{\mathcal{I}}(v)$ is the largest demand (note that for k-MEDIAN, k-MEANS, or FACILITY LOCATION we typically have $X = 1$).

Theorem 2. *For any $\varepsilon > 0$, a $(1+\varepsilon)$-approximation for k-*CLUSTERING^q *and* FACILITY LOCATION^q *can be computed in $(nX)^{(hq/\varepsilon)^{O(q)}}$ time on graphs of highway dimension h.*

In particular, this algorithm is much faster than the quasi-polynomial time approximation scheme of Feldmann et al. [17] for k-MEDIAN or FACILITY LOCATION. The runtime of our algorithm also significantly improves over the exponential dependence on k in the approximation schemes of Becker et al. [8] and Braverman et al. [10] for k-MEDIAN.

It has so far been open whether these clustering problems are NP-hard on graphs of constant highway dimension. We complement our main theorem by showing that they are NP-hard even for the smallest possible highway dimension. This answers an open problem given in [17]. Here the *uniform* FACILITY LOCATION^q problem has unit opening costs for all facilities.

Theorem 3. *The k-*CLUSTERING^q *and uniform* FACILITY LOCATION^q *problems are NP-hard on graphs of highway dimension* 1*.*

1.2 Related work

On clustering problems. The problems we focus on in this article are known to be APX-hard in general metric spaces (see e.g. [6, 20, 23]). The current best polynomial-time algorithm for FACILITY LOCATION achieves a 1.488-approximation [25], while the best approximation factor is 2.67 for k-MEDIAN [11] and 6.357 for k-MEANS [4].

When restricting the inputs, a near-linear time approximation scheme for doubling metrics was developed in [12]; we will discuss the close relations between our work and this one in Section 1.3. Local search techniques also yield a PTAS in metrics arising from classes of minor-free graphs and metrics with bounded doubling dimension [13, 19], and a $\Theta(q)$-approximation for the k-CLUSTERING^q problem in general metric spaces [22].

Another technique for dealing with clustering problems is to compute *coresets*, which are compressed representations of the input. An ε-coreset is a weighted set of points such that for every set of centers, the cost for the original set of points is within a $(1+\varepsilon)$-factor of the cost for the coreset. Braverman et al. [10] recently proved that graphs with highway dimension h admit coreset of size $\widetilde{O}((k+h)^{O(1/\varepsilon)})$. This enables to compute a $(1+\varepsilon)$-approximation by enumerating all possible solutions of the coreset. However, this coreset does not have small highway dimension,[1] and thus cannot be used to boost our algorithms.

On highway dimension. The highway dimension was originally defined by Abraham et al. [3], who specifically chose balls of radius $4r$ in the Definition 1. Since the original definition in [3], several other definitions have been proposed. In particular, Feldmann et al. [17] proved that when choosing a radius cr in Definition 1 for any constant c strictly larger than 4, it is possible to exploit the structure of graphs with constant highway dimension in order to obtain a QPTAS for problems such as TSP, FACILITY LOCATION, and STEINER TREE. As Abraham et al. [3] point

[1] Indeed, a subset of a metric with small highway dimension does not necessarily have small highway dimension as well: think of a star metric on which the center is removed.

out, the choice of the constant is somewhat arbitrary, and we use the above definition so that we may exploit the structural insights of [17] for our algorithm. These structural properties were also leveraged by Becker et al. [8] who gave a PTAS for the Bounded-Capacity Vehicle Routing problem, and a parameterized approximation scheme for the k-Center problem (which is essentially k-Clustering^q with $q = \infty$) and k-Median. On the lower-bound side, Disser et al. [15] showed that Steiner Tree and TSP are weakly NP-hard even when the highway dimension is 1, i.e., each of them is NP-hard but an FPTAS exists.

It is worth mentioning that further definitions of the highway dimension exist (for a detailed discussion see Appendix A and [9, 17]). In particular, for a more general definition of the highway dimension than the one of Definition 1, Feldmann [16] gave a parameterized 3/2-approximation algorithm with runtime $2^{O(kh \log h)} n^{O(1)}$ for k-Center.

1.3 Our techniques

To obtain Theorem 2, we rely on the framework recently developed by Cohen-Addad et al. [12] for doubling metrics. More precisely, they show that the *split-tree decomposition* of Talwar [26] has some interesting properties, and exploit them to design their algorithm. Our main contribution is to provide a decomposition with similar properties in graphs with constant highway dimension. This is done relying on some structural properties of such graphs presented by Feldmann et al. [17]. We start by giving an outline of the algorithm from [12], and then explain how to carry the results over to the highway dimension setting.

On doubling metrics. The starting point of many approximation algorithms for doubling metrics is a decomposition of the metric, as presented in the following lemma taken from [12]. A *hierarchical decomposition* $\mathcal{D}$ of a metric (V, dist) is a sequence of partitions $\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_\lambda$ of V, where $\mathcal{B}_i$ refines $\mathcal{B}_{i+1}$, i.e., every part $B \in \mathcal{B}_i$ is contained in some part of $\mathcal{B}_{i+1}$. Moreover, in $\mathcal{B}_0$ every part contains a singleton vertex of V, while $\mathcal{B}_\lambda$ contains only one part, namely V. For a point $v \in V$ and a radius $r > 0$, we say that the ball $\beta_v(r)$ is *cut at level i* if i is the largest integer for which the ball $\beta_v(r)$ is not contained in a single part of $\mathcal{B}_i$. For any subset $W \subseteq V$ of vertices we define $\lambda(W) = \lceil \log_2 \text{diam}(W) \rceil$, where $\text{diam}(W) = \max_{u,v \in W} \text{dist}(u, v)$ is the diameter of W.

Lemma 4 (Reformulation of [7, 26] as found in [12][2])**.** *For any metric (V, dist) of doubling dimension d and any $\rho > 0$, there exists a polynomial-time computable randomized hierarchical decomposition $\mathcal{D} = \{\mathcal{B}_0, \ldots, \mathcal{B}_{\lambda(V)}\}$ such that the diameter of each part $B \in \mathcal{B}_i$ is at most 2^{i+1}, and:*

1. ***Scaling probability**: for any $v \in V$, radius r, and level i, we have $\Pr[\mathcal{D} \text{ cuts } \beta_v(r) \text{ at level } i] \leq 2^{O(d)} \cdot r/2^i$.*
2. ***Portal set:** every part $B \in \mathcal{B}_i$ where $\mathcal{B}_i \in \mathcal{D}$ comes with a set of* portals *$P_B \subseteq B$ that is*
   (a) ***concise:** the size of the portal set is bounded by $|P_B| \leq 1/\rho^d$, and*
   (b) ***precise:** for every node $u \in B$ there is a portal $p \in P_B$ with $\text{dist}(u, p) \leq \rho 2^{i+1}$.*

We briefly sketch the standard use of this decomposition. For clustering problems, one can show that there exists a *portal-respecting solution* with near-optimal cost (see Talwar [26]). In this structured solution, each client connects to a facility via a *portal-respecting path* that enters and leaves any part B of $\mathcal{D}$ only through a node of the portal set P_B. Those portals therefore act as

[2] We remark that in [12] the preciseness of Lemma 4 was expressed akin to the weaker property found in Lemma 5, which however would not lead to a near-linear time approximation scheme as claimed in [12], but rather a PTAS as shown in this work. This can however easily be alleviated for [12] by using the stronger preciseness as stated here in Lemma 4.

separators of the metric. A standard dynamic program approach can then compute the best portal respecting solution.

To ensure that there is a portal-respecting solution with near-optimal cost, one uses the preciseness property of the portal set: the distortion (i.e., the overhead) of connecting a client c with a facility f through portals instead of directly is bounded as follows. Let i be the level at which $\mathcal{D}$ *cuts* c and f, meaning that i is the maximum integer for which c and f lie in different parts of $\mathcal{B}_i$. At every level $j \leq i$ the portal-respecting path uses an edge to the closest portal on this level, and thus incurs a distortion of $\rho 2^{j+1}$. Hence the total distortion is $\sum_{j \leq i} O(\rho 2^j) = O(\rho 2^i)$. Now, the scaling probability of the decomposition ensures that c and f are cut at level i with probability at most $2^{O(d)} \operatorname{dist}(c, f)/2^i$. Hence combining those two bounds over all levels ensures that, in expectation, the distortion between c and f is $2^{O(d)} \operatorname{dist}(c, f) \cdot \rho\lambda(V)$. Using a standard pre-processing technique (see e.g. [17]) we may reduce $\lambda(V)$ to $O(\log(n/\varepsilon))$ when aiming for a $(1+\varepsilon)$-approximation. Hence choosing $\rho = \frac{\varepsilon}{2^{O(d)} \log n}$ gives a distortion of $\varepsilon \cdot \operatorname{dist}(c, f)$. Summing over all clients proves that there exists a near-optimal portal-respecting solution.

The issue with this approach is that by the conciseness property, the number of needed portals is $2^{O(d^2)} \log^d n/\varepsilon^d$, and the dynamic program has a runtime that is exponential in this number. Thus the time complexity is quasipolynomial. The novelty of [12] is to show how to reduce the number of portals to a constant. The idea is to reduce the number of levels on which a client can be cut from its facility.

For this, Cohen-Addad et al. [12] present a processing step of the instance that helps dealing with clients cut from their facility at a high level (see Section 2 for formal definitions and lemmas). Roughly speaking, their algorithm computes a constant factor approximation L of k-Clustering^q or Facility Location^q, and a client c is called *badly-cut* if $\mathcal{D}$ cuts it from its closest facility of L at a level larger than $\log(\operatorname{dist}(c, L)/\varepsilon) + \tau(\varepsilon, q, d)$ for some function τ. Every badly-cut client is moved to its closest facility of L. It is then shown that this new instance $\mathcal{I}_\mathcal{D}$ has *small distortion*, which essentially means that any solution to $\mathcal{I}_\mathcal{D}$ can be converted to a solution of the original instance $\mathcal{I}$ while only losing a $(1+\varepsilon)$-factor in quality. In this instance $\mathcal{I}_\mathcal{D}$ all clients are cut from their closest facility of L at some level between $\log(\operatorname{dist}(c, L)/2)$ and $\log(\operatorname{dist}(c, L)/\varepsilon) + \tau(\varepsilon, q, d)$ (where the lower bound holds because two vertices at distance d cannot be in the same part of any level smaller than $\log(d/2)$ due to the diameter bound of each part given by Lemma 4). Using this property, it can be shown that c and its closest center in the optimal solution are also cut at a level in that range. As there are only $O(\log(1/\varepsilon)) + \tau(\varepsilon, q, d)$ levels in this range, by the previous argument, the number of portals is now independent of n.

On highway dimension. The above arguments for doubling metrics hold thanks to Lemma 4. In this work, we show how to construct a similar decomposition for low highway dimension:

Lemma 5. *Given a shortest-path metric (V, dist) of a graph with highway dimension h, a subset $W \subseteq V$, and $\rho > 0$, there exists a polynomial-time computable randomized hierarchical decomposition $\mathcal{D} = \{\mathcal{B}_0, \ldots, \mathcal{B}_{\lambda(W)}\}$ of W such that the diameter of each part $B \in \mathcal{B}_i$ is at most 2^{i+5}, and:*

1. ***Scaling probability**: for any $v \in V$, radius r, and level i, we have* $\Pr[\mathcal{D} \text{ cuts } \beta_v(r) \text{ at level } i] \leq \sigma \cdot r/2^i$*, where* $\sigma = (h \log(1/\rho))^{O(1)}$.
2. ***Interface**: for any $B \in \mathcal{B}_i$ on level $i \geq 1$ there exists an interface $I_B \subseteq V$, which is*
   - *(a)* ***concise**: $|I_B| \leq (h/\rho)^{O(1)}$, and*
   - *(b)* ***precise**: for any $u, v \in B$ such that u and v are cut by $\mathcal{D}$ at level $i-1$, there exists $p \in I_B$ with* $\operatorname{dist}(u, p) + \operatorname{dist}(p, v) \leq \operatorname{dist}(u, v) + 68 \cdot \rho 2^i$.

Our construction relies on the *town decomposition* from [17], which is a laminar family with the following properties (see Section 2 for formal definitions and lemmas). If $\mathcal{T}$ is a town decomposition of a metric (V, dist), then every $T \in \mathcal{T}$ is a subset of V and is called a *town*, every vertex $v \in V$ is contained in at least one town, and also the whole set V is a town of $\mathcal{T}$. Similar to hierarchical decompositions, the laminar family $\mathcal{T}$ thus decomposes V. If the metric is given by a graph of highway dimension h, for a given $\rho > 0$ every town $T \in \mathcal{T}$ has a set X_T of *hubs* with doubling dimension $O(\log(h \log(1/\rho)))$, such that for any two vertices u and v in different child towns of T, there is a hub $x \in X_T$ such that $\mathrm{dist}(u, x) + \mathrm{dist}(x, v) \leq (1 + 2\rho) \cdot \mathrm{dist}(u, v)$.

This hub set X_T is similar to the portal set of Lemma 4, but has some fundamental differences: the first one is that the town decomposition is deterministic, and so it may happen that a client and its facility are cut at a very high level — something that happens only with tiny probability in the doubling setting thanks to the scaling probability. Another main difference is that the size of X_T might be unbounded. As a consequence, it cannot be directly used as a portal set in a dynamic program. To deal with this, we combine the town decomposition with a hierarchical decomposition of each set X_T according to Lemma 4, to build an *interface* as stated in Lemma 5.

A further notable difference to portals is that the preciseness property of the resulting interface is weaker. In particular, while there is a portal close to each vertex of a part, the hubs (and consequently the interface points) can be far from some vertices as long as they lie close to the shortest path to other vertices. This means that no analogue to near-optimal portal-respecting paths exist (see Appendix A). Instead, when connecting a client c with a facility f we need to use the interface point of I_B, provided by the preciseness property of Lemma 5, that lies close to the shortest path between c and f for the lowest level part B containing both c and f. This shifts the perspective from externally connecting vertices of a part to vertices outside a part, as done for portals, to internally connecting vertices of parts, as done here.

As a consequence, we develop a dynamic program, which follows more or less standard techniques as for instance given in [5, 24], but needs to handle the weaker preciseness property of the interface. The main idea is to guess the distances from interface points to facilities while recursing on the decomposition $\mathcal{D}$ of Lemma 5. The runtime of this algorithm is thus exponential in the number of interface points. Thanks to the techniques developed by Cohen-Addad et al. [12] as described above, we can assume that this number is constant for clustering problems. However, due to the shifted perspective towards internally connecting vertices of parts, the runtime of the dynamic program also is exponential in the *total* number of levels. It can be shown though that it suffices to compute a solution on a carefully chosen subset W of the metric for which only a logarithmic number of levels of the decomposition need to be considered. Thus the overall runtime is bounded by some constant raised to a logarithm, which is polynomial.

1.4 Outline

After defining the concepts we use and stating various structural lemmas in Section 2, we show how to incorporate our decomposition into the framework of [12]. The proof of Lemma 5 is then presented in Section 3. The formal algorithm can be found in Section 4. We conclude with the hardness proof of Theorem 3 in Section 5.

2 Preliminaries

On doubling metrics. The *doubling dimension* of a metric is the smallest integer d such that for any $r > 0$ and $v \in V$, the ball $\beta_v(2r)$ of radius $2r$ around v can be covered by at most 2^d balls of half the radius r. A doubling metric is a metric space where the doubling dimension is constant. In those spaces one can show the existence of small *nets*:

Definition 6. *A* δ-net *of a metric* (V, dist) *is a subset of nodes* $N \subseteq V$ *with the property that every node in* V *is at distance at most* δ *from a net point of* N*, and each pair of net points of* N *are at distance more than* δ*.*

Note that a simple greedy algorithm can compute a δ-net for any given metric in polynomial time. In low doubling metrics these nets have the following useful properties, as shown by Gupta et al. [21].

Lemma 7 ([21])**.** *Let* (V, dist) *be a metric space with doubling dimension* d*. If its diameter is* D*, and* N *is a* δ*-net of* V*, then* $|N| \leq 2^{d \cdot \lceil \log_2(D/\delta) \rceil}$*. Moreover, any subset* $W \subseteq V$ *has doubling dimension at most* $2d$*.*

On highway dimension. For simplicity we will set $c = 8$ in Definition 1 throughout this paper, even if all claimed results are also true for other values of c. When we refer to a metric as having highway dimension h, we mean that it is the shortest-path metric of a graph of highway dimension h. A *laminar family* of V is a set system with universe V in which no two sets cross, i.e., any two sets are either disjoint or one set is contained in the other. This naturally gives rise to a rooted tree structure on the sets, and we thus refer to proper subsets of a set as its *descendants* and to inclusion-wise maximal proper subsets as its *children*. The main result we will use about highway dimension is the existence of the following decomposition:

Theorem 8 ([17])**.** *Given a shortest-path metric* (V, dist) *of highway dimension* h*, and* $\rho > 0$*, there exists a polynomial-time computable deterministic laminar family* $\mathcal{T}$ *of* V*, called the* town decomposition*, where every set* $T \in \mathcal{T}$ *is called a* town*. For every vertex* $v \in V$ *there is a singleton town* $\{v\} \in \mathcal{T}$*, and also* $V \in \mathcal{T}$*. Every town* T *has a set of* hubs[3] $X_T \subseteq T$ *with the following properties:*

a. ***doubling****: the doubling dimension of* X_T *is* $d = O(\log(h \log(1/\rho)))$*, and*
b. ***precise****: for any two vertices* u *and* v *in different child towns of* T*, there is a vertex* $x \in X_T$ *such that* $\mathrm{dist}(u, x) + \mathrm{dist}(x, v) \leq (1 + 2\rho) \cdot \mathrm{dist}(u, v)$*.*

The town decomposition behaves differently from those in Lemmas 4 and 5 in several ways. The main properties we will need here are given by the following lemma. Given a sequence of towns $T_0, \ldots, T_g$ of the towns decomposition such that T_ℓ is a child town of $T_{\ell-1}$ for each $\ell \in \{1, \ldots, g\}$, we call T_g an g^{th}*-generation descendant* of T_0. In particular, a child town is a 1^{st}-generation descendant.[4] The given property on these descendants is implicit in [17] and we give a proof outline in Appendix B.

Lemma 9 ([17])**.** *For any* $T \in \mathcal{T}$ *we have* $\mathrm{diam}(T) < \mathrm{dist}(T, V \setminus T)$*. Furthermore, if* T' *is an* g^{th}*-generation descendant town of* T*, then* $\mathrm{diam}(T') < \mathrm{diam}(T)/2^{g-1}$*.*

On how to incorporate our decomposition into the framework of [12]. Assume we are given an instance $\mathcal{I}$ of k-Clustering^q or Facility Location^q on some metric (V, dist), together with a hierarchical decomposition $\mathcal{D}$ of the metric with the properties listed in Lemma 5. We start by defining the *badly cut* clients. In the following, we fix an optimal solution OPT and an approximate solution L, and we define $\tau(\varepsilon, q, \sigma) = \log_2(\sigma(q+1)^q/\varepsilon^{q+1})$.

[3] called *approximate core hubs* in [17].

[4] We note that in the conference version of this paper [18] it was erroneously claimed that for any child town T' of a town T we have $\mathrm{diam}(T') < \mathrm{diam}(T)/2$.

Definition 10 (badly cut [12]). *Let (V, dist) be a metric of an instance $\mathcal{I}$ of k-*Clustering^q *or* Facility Location^q*, $\mathcal{D}$ be a hierarchical decomposition of the metric with scaling probability factor σ, and $\varepsilon > 0$. If L_v is the distance from v to the closest facility of an approximate solution L to $\mathcal{I}$, then a client c is* badly cut w.r.t. *$\mathcal{D}$ if the ball $\beta_c(3L_c/\varepsilon)$ is cut at some level i greater than $\log_2(3L_c/\varepsilon) + \tau(\varepsilon, q, \sigma)$.*

Similarly, if OPT_v is the distance from v to the closest facility of the optimum solution OPT *of $\mathcal{I}$, then a facility $f \in L$ is* badly cut w.r.t. *$\mathcal{D}$ if $\beta_f(3\mathrm{OPT}_f)$ is cut at some level i greater than $\log_2(3\mathrm{OPT}_f) + \tau(\varepsilon, q, \sigma)$.*

Given an instance $\mathcal{I}$ of k-Clustering^q or Facility Location^q and a decomposition $\mathcal{D}$ of the metric, a new instance $\mathcal{I}_\mathcal{D}$ is computed to get rid of badly cut clients. The instance $\mathcal{I}_\mathcal{D}$ is built from $\mathcal{I}$ by moving clients that are badly cut w.r.t. $\mathcal{D}$ to their closest facility in L.[5] For any client c of $\mathcal{I}_\mathcal{D}$ we denote by $\tilde{c}$ the original position of this client in $\mathcal{I}$, i.e., if $\tilde{c}$ is a badly cut client of $\mathcal{I}$ then $c = L(\tilde{c})$ and otherwise $c = \tilde{c}$. The set F of potential centers in unchanged, and thus any solution of $\mathcal{I}$ is a solution of $\mathcal{I}_\mathcal{D}$, and vice versa. Note that $\mathcal{I}_\mathcal{D}$ does not contain any badly cut client w.r.t. $\mathcal{D}$, and that the definition of $\mathcal{I}_\mathcal{D}$ depends on the randomness of $\mathcal{D}$.

To describe the properties we obtain for the new instance, given a solution S to any instance $\mathcal{I}_0$ of k-Clustering^q or Facility Location^q, we define $\mathrm{cost}_{\mathcal{I}_0}(S) = \sum_{v \in V} \chi_{\mathcal{I}_0}(v) \cdot \mathrm{dist}(v, S)^q$ to be the cost incurred by only the distances to the facilities. Note that for k-Clustering^q this coincides with the objective function, while for Facility Location^q we need to also add the facility opening costs to $\mathrm{cost}_{\mathcal{I}_0}(S)$ to obtain the objective function. Given some $\varepsilon > 0$ and the computed instance $\mathcal{I}_\mathcal{D}$ from $\mathcal{I}$, we define

$$\nu_{\mathcal{I}_\mathcal{D}} = \max_{\text{solution } S} \left\{ \mathrm{cost}_\mathcal{I}(S) - (1 + 2\varepsilon) \mathrm{cost}_{\mathcal{I}_\mathcal{D}}(S) \ , \ (1 - 2\varepsilon) \mathrm{cost}_{\mathcal{I}_\mathcal{D}}(S) - \mathrm{cost}_\mathcal{I}(S) \right\}.$$

If $B_\mathcal{D}$ denotes the set of badly cut facilities (w.r.t $\mathcal{D}$) of the solution L to $\mathcal{I}$ from which instance $\mathcal{I}_\mathcal{D}$ is constructed, we say that $\mathcal{I}_\mathcal{D}$ has *small distortion w.r.t. $\mathcal{I}$* if $\nu_{\mathcal{I}_\mathcal{D}} \leq \varepsilon \, \mathrm{cost}_\mathcal{I}(L)$, and there exists a *witness solution* $\hat{S} \subseteq F$ that contains $B_\mathcal{D}$ and for which $\mathrm{cost}_{\mathcal{I}_\mathcal{D}}(\hat{S}) \leq (1 + O(\varepsilon)) \mathrm{cost}_\mathcal{I}(\mathrm{OPT}) + O(\varepsilon) \mathrm{cost}_\mathcal{I}(L)$. Moreover, in the case of Facility Location^q, $\hat{S} = \mathrm{OPT} \cup B_\mathcal{D}$ and $\sum_{f \in B_\mathcal{D}} w_f \leq \varepsilon \cdot \sum_{f \in L} w_f$.

Based on these definitions, we now state the main tool we use from [12], and which exploits the scaling probability of our decomposition in Lemma 5 to obtain the required structure.

Lemma 11 ([12]). *Let (V, dist) be a metric, and $\mathcal{D}$ be a randomized hierarchical decomposition of (V, dist) with scaling probability factor σ. Let $\mathcal{I}$ be an instance of k-*Clustering^q *or* Facility Location^q *on (V, dist), with optimum solution* OPT *and approximate solution L. For any (sufficiently small) $\varepsilon > 0$, with probability at least $1 - \varepsilon$ (over $\mathcal{D}$), the instance $\mathcal{I}_\mathcal{D}$ constructed from $\mathcal{I}$ and L as descibed above has small distortion with a witness solution $\hat{S}$. Furthermore, every client c of $\mathcal{I}_\mathcal{D}$ is cut by $\mathcal{D}$ from its closest facility in $\hat{S}$ at level at most $\log_2(3L_{\tilde{c}}/\varepsilon + 4\mathrm{OPT}_{\tilde{c}}) + \tau(\varepsilon, q, \sigma)$, where $\tilde{c}$ is the original position of c in $\mathcal{I}$.*

As a consequence of Lemma 11, a dynamic program can compute a solution recursively on the parts of $\mathcal{D}$ in polynomial time, as sketched in Section 1.3 and detailed in Section 4.

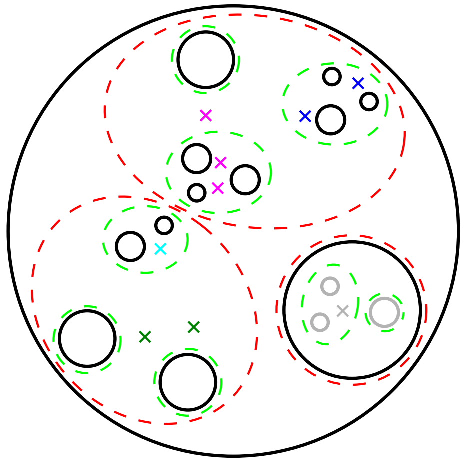

Figure 1: A town and its child towns (black circles). The hubs (crosses) are decomposed by $\mathcal{X}_T$, indicated by different colours (note that in reality the hubs are contained in child towns, but are depicted separately here for clarity). Parts $B \in \mathcal{B}_{i+1}$ (red dashed) are decomposed into parts on level i (green dashed). Parts of $\mathcal{B}_i$ can lie in different towns (e.g., the child town of T containing grandchild towns in grey).

3 Decomposing the graph

This section is dedicated to the proof of Lemma 5. For this we first fix a town decomposition $\mathcal{T}$ of the input graph, as given by Theorem 8 assuming (for technical reasons) that $\rho \leq 1/2$. The general idea to construct a hierarchical decomposition $\mathcal{D}$ is as follows. For doubling metrics, to decompose a part at level i, it is enough to pick a random diameter $\delta \in [2^{i-2}, 2^{i-1})$ and divide the part into child parts of diameter δ. This is not doable in the highway dimension setting: if one wishes to decompose a town T, it cannot divide any of the child towns, since it is not possible to use the hubs X_T of T to approximate paths inside one of the child towns. The high-level picture of our decomposition is therefore as follow. To decompose a town at level i, we group the "small" child towns randomly (as in the doubling decomposition), and put every other child town in its own subpart. As we will see, this turns out to be enough.

In order to decompose a town T, we need the following definitions. For each child town T' of T we identify the *connecting hub* $x \in X_T$, which is some fixed closest hub of X_T to T', breaking ties arbitrarily. Moreover, given a hierarchical decomposition $\mathcal{X}_T = \{\mathcal{U}_0, \dots, \mathcal{U}_{\lambda(X_T)}\}$ of X_T, we define for every i the *connecting i-cluster* of a child town T' of T to be the set $U \in \mathcal{U}_\ell$ on level $\ell = \min\{i, \lambda(X_T)\}$ containing the connecting hub of T'. For a given subset $W \subseteq V$ we then follow the steps below, after choosing μ from the interval $(0, 1]$ uniformly at random (cf. Fig. 1):

1. For each town $T \in \mathcal{T}$, we apply Lemma 4 to find a randomized hierarchical decomposition $\mathcal{X}_T = \{\mathcal{U}_0, \dots, \mathcal{U}_{\lambda(X_T)}\}$ of the hubs X_T of T.
2. In this step we fix a town $T \in \mathcal{T}$. Using $\mathcal{X}_T$, we define a randomized partial decomposition of $T \cap W$ as follows. For any i and $U \in \mathcal{U}_{\min\{i,\lambda(X_T)\}}$, let the set $A_i^U \subseteq T \cap W$ be the union of all sets $T' \cap W$ where T' is a child town of T with the following two properties:
   (a) U is the connecting i-cluster of T', and
   (b) $\mathrm{dist}(T', V \setminus T') \leq \mu 2^i$.

   In particular, A_i^U contains towns somewhat close to U due to (a) and with small diameter

[5] More concretely, let $\chi_{\mathcal{I}}$ and $\chi_{\mathcal{I}_\mathcal{D}}$ be the demand functions of $\mathcal{I}$ and $\mathcal{I}_\mathcal{D}$, respectively. Initially we let $\mathcal{I}_\mathcal{D}$ be a copy of $\mathcal{I}$, so that in particular $\chi_{\mathcal{I}_\mathcal{D}} = \chi_{\mathcal{I}}$. Then, for each client c of $\mathcal{I}$ that is badly cut in L w.r.t. $\mathcal{D}$, if $L(c)$ denotes the closest facility of L to c, in $\mathcal{I}_\mathcal{D}$ we set $\chi_{\mathcal{I}_\mathcal{D}}(c) = 0$ and increase $\chi_{\mathcal{I}_\mathcal{D}}(L(c))$ by the value of $\chi_{\mathcal{I}}(c)$ in $\mathcal{I}$.

due to (b) and Lemma 9. We let $\mathcal{A}_i^T$ be the set containing each non-empty A_i^U.

3. Now, the hierarchical decomposition $\mathcal{D} = \{\mathcal{B}_0, \dots, \mathcal{B}_{\lambda(W)}\}$ of W can be constructed inductively as follows. At the highest level $\lambda(W)$ of $\mathcal{D}$, W is partitioned into a single set: $\mathcal{B}_{\lambda(W)} = \{W\}$. To decompose a part $B \in \mathcal{B}_{i+1}$ at level $i+1$, we do the following. Let $T \in \mathcal{T}$ be the inclusion-wise minimal town for which $B \subseteq T$. The "small" child towns of T lying inside B are grouped according to step 2 (note that $\operatorname{dist}(T', V \setminus T')$ also bounds the diameter of T' by Lemma 9), and the other ones form individual subparts (note though that these remaining child towns may also have small diameter and thus it would be misleading to call them "big"). More formally, the set $\mathcal{B}_i$ contains every part $A \in \mathcal{A}_i^T$ for which $A \subseteq B$, and also every set $T' \cap W$, where T' is a child town of T for which $T' \cap W \subseteq B$ and $T' \cap W$ was not covered by the previously added parts of $\mathcal{A}_i^T$, i.e., $T' \cap W \cap A = \emptyset$ for every $A \in \mathcal{A}_i^T$.

To prove that the constructed decomposition $\mathcal{D}$ has the desired properties – i.e. that it is indeed a hierarchical decomposition with parts of bounded diameter and small scaling probability – we begin with some auxiliary lemmas, of which the first one bounds the distance of a town to its connecting hub.

Lemma 12. *If T' is a child town of T with connecting hub $x \in X_T$, then we have* $\operatorname{dist}(x, T') \leq (1+2\rho)\operatorname{dist}(T', V \setminus T')$.

Proof. Let T'' be the closest sibling town to T', and let $u \in T'$ and $v \in T''$ be the vertices defining the distance from T' to T'', i.e., $\operatorname{dist}(u,v) = \operatorname{dist}(T', T'') = \operatorname{dist}(T', V \setminus T')$. By Theorem 8, there is a hub $y \in X_T$ for which $\operatorname{dist}(u,y) + \operatorname{dist}(y,v) \leq (1+2\rho) \cdot \operatorname{dist}(u,v) = (1+2\rho) \cdot \operatorname{dist}(T', V \setminus T')$. This implies $\operatorname{dist}(y, T') \leq \operatorname{dist}(u,y) \leq (1+2\rho) \cdot \operatorname{dist}(T', V \setminus T')$. Since the connecting hub x of T' is at least as close to T' as y, the claim follows. □

Based on the above lemma, we next prove the key property that the diameter of any part of $\mathcal{B}_i \in \mathcal{D}$ is bounded.

Lemma 13. *If $\rho \leq 1/2$, then the diameter of any part of $\mathcal{B}_i \in \mathcal{D}$ is less than 2^{i+5}.*

Proof. On the highest level $\lambda(W)$ of $\mathcal{D}$ the only part of $\mathcal{B}_{\lambda(W)}$ is W itself. As $\lambda(W) = \lceil \log_2 \operatorname{diam}(W) \rceil$ we get $\operatorname{diam}(W) \leq 2^{\lambda(W)+1}$, as required.

For any level $i < \lambda(W)$, a set in $\mathcal{B}_i$ is either equal to a set $A \in \mathcal{A}_i^T$ for some town $T \in \mathcal{T}$ or it is equal to some set $T' \cap W$ for a child town T' of T. In the former case, the set A is a set A_i^U for some cluster $U \in \mathcal{U}_\ell$ where $\ell = \min\{i, \lambda(X_T)\}$ and $\mathcal{U}_\ell \in \mathcal{X}_T$. The set A_i^U contains the union of sets $T' \cap W$ for child towns T' of T, for which their connecting hubs lie in U and $\operatorname{dist}(T', V \setminus T') \leq \mu 2^i \leq 2^i$, as $\mu \leq 1$. Thus from Lemma 12 we get $\operatorname{dist}(U, T') \leq (1+2\rho)2^i$, and by Lemma 9 we have $\operatorname{diam}(T') < \operatorname{dist}(T', V \setminus T') \leq 2^i$. The cluster U has diameter at most 2^{i+1} by Lemma 4, since it is part of the hierarchical decomposition $\mathcal{X}_T$ and lies on level $\ell \leq i$. Let u and v be the vertices of A_i^U defining the diameter of A_i^U, i.e., $\operatorname{dist}(u,v) = \operatorname{diam}(A_i^U)$. We may reach v from u by first crossing the child town T' that u lies in, then passing over to U, then crossing U, after which we pass over to the child town T'' containing v, and finally crossing this child town as well to reach v. Hence, assuming that $\rho \leq 1/2$ the diameter of A_i^U is bounded by

$$\begin{aligned}\operatorname{dist}(u,v) &\leq \operatorname{diam}(T') + \operatorname{dist}(U,T') + \operatorname{diam}(U) + \operatorname{dist}(U,T'') + \operatorname{diam}(T'') \\ &< 2 \cdot 2^i + 2 \cdot (1+2\rho)2^i + 2^{i+1} = (6+4\rho)2^i \leq 2^{i+3}\end{aligned}$$

Now consider the other case, when a set $B \in \mathcal{B}_i$ on level $i < \lambda(W)$ is equal to some set $T' \cap W$ for a child town T' of a town T. For such a child town T' there is no enforced upper bound on the distance to other child towns as before, and thus it is necessary to be more careful to bound the diameter of the part. Starting with $B = B_i$, let $B_i \subseteq B_{i+1} \subseteq \ldots \subseteq B_j$ be the longest chain of parts of increasing levels that are of the same type as B. More concretely, for every $\ell \in \{i, i+1, \ldots, j\}$ we have $B_\ell \in \mathcal{B}_\ell$ and B_ℓ is equal to some set $T'_\ell \cap W$ for a child town T'_ℓ of the inclusion-wise minimal town T_ℓ containing $B_{\ell+1}$. Note that in particular $j < \lambda(W)$. As we chose the longest such chain, on the next level $j+1$ there is no such set containing B_j, which means that the set $B_{j+1} \in \mathcal{B}_{j+1}$ for which $B_j \subseteq B_{j+1}$ is either equal to a set $A \in \mathcal{A}_{j+1}^{T_{j+1}}$ for some town T_{j+1}, or $j+1 = \lambda(W)$. In either case, from above we get $\operatorname{diam}(B_{j+1}) \leq 2^{j+4}$.

Note that for any $\ell \in \{i, i+1, \ldots, j-1\}$, since $B_{\ell+1} = T'_{\ell+1} \cap W$ implies $B_{\ell+1} \subseteq T'_{\ell+1}$, while T_ℓ is the inclusion-wise minimal town containing $B_{\ell+1}$, we have $T_\ell \subseteq T'_{\ell+1}$. Now, as T'_ℓ is a child town of T_ℓ, we get that T'_ℓ is a descendant of $T'_{\ell+1}$. This means that T'_i is a g^{th}-generation descendant of T'_j for some $g \geq j-i$, and from Lemma 9 we get $\operatorname{diam}(T'_i) \leq \operatorname{diam}(T'_j)/2^{j-i-1}$. As $B = B_i \subseteq T'_i$ we have $\operatorname{diam}(B) \leq \operatorname{diam}(T'_i)$. Since T_j is the inclusion-wise minimal town containing B_{j+1}, the latter set contains vertices of at least two child towns of T_j. One of these child towns is T'_j, since $B_j = T'_j \cap W$ and $B_j \subseteq B_{j+1}$ by construction of the decomposition. In particular, B_{j+1} both contains vertices inside and outside of T'_j, and so $\operatorname{dist}(T'_j, V \setminus T'_j) \leq \operatorname{diam}(B_{j+1})$. By Lemma 9 we know that $\operatorname{diam}(T'_j) < \operatorname{dist}(T'_j, V \setminus T'_j)$, and putting all these inequalities together we obtain

$$\begin{aligned}\operatorname{diam}(B) \leq \operatorname{diam}(T'_i) \leq \operatorname{diam}(T'_j)/2^{j-i-1} < \operatorname{dist}(T'_j, V \setminus T'_j)/2^{j-i-1} \\ \leq \operatorname{diam}(B_{j+1})/2^{j-i-1} \leq 2^{j+4}/2^{j-i-1} = 2^{i+5}. \quad \square\end{aligned}$$

Using Lemma 13 it is not hard to prove the correctness of $\mathcal{D}$, which we turn to next.

Lemma 14. *The tuple $\mathcal{D} = \{\mathcal{B}_0, \ldots, \mathcal{B}_{\lambda(W)}\}$ is a hierarchical decomposition of W.*

Proof. We first prove that for a part $B \in \mathcal{B}_i$ included in town T, part B can be partitioned into unions of sets $T' \cap W$ for child towns T' of T. Indeed, either $B = T \cap W$, and properties of the town decomposition ensure that B can be partitioned in this way, or $B \in \mathcal{A}_i^T$. By construction of $\mathcal{A}_i^T$, in the latter case part B is also the union of sets $T' \cap W$ for child towns T' of T.

Now, step (3) of the construction decomposes B into groups of child towns restricted to W, and so B is partitioned by $\mathcal{B}_{i-1}$. Moreover, since $\mathcal{B}_{\lambda(W)} = \{W\}$, by induction each $\mathcal{B}_i$ is a partition of W. That concludes the proof. ☐

We now turn to proving the properties of Lemma 5, starting with the scaling probability.

Lemma 15. *The decomposition $\mathcal{D}$ has scaling probability factor $\sigma = (h \log(1/\rho))^{O(1)}$.*

Proof. To prove the claim, we need to prove that for any $v \in W$, radius r, and level i, the probability that $\mathcal{D}$ cuts the ball $\beta_v(r)$ at level i is at most $(h \log(1/\rho))^{O(1)} \cdot r/2^i$. If $\mathcal{D}$ cuts $\beta_v(r)$ at level i, it means that $\beta_v(r)$ is fully contained in a part at level $i+1$. Let $T \in \mathcal{T}$ be the inclusion-wise minimal town containing that part. There are two cases to consider: either $\beta_v(r)$ is cut by "small" parts, i.e. there exist two distinct parts $A_1, A_2 \in \mathcal{A}_i^T$ such that $v \in A_1$ and $u \in A_2$ for some $u \in W \cap \beta_v(r)$, or not.

We start with the latter case, when $\beta_v(r)$ is not cut by small parts. If $\mathcal{D}$ cuts the ball at level i, there are distinct parts $B, B' \in \mathcal{B}_i$ such that $v \in B$ and $u \in B'$ for some $u \in W \cap \beta_v(r)$. Assume w.l.o.g. that $B \notin \mathcal{A}_i^T$ (which is possible to assume since $\beta_v(r)$ is not cut by small parts). By construction of the decomposition, there must be a child town T' of T, for which $B = T' \cap W$ and $\operatorname{dist}(T', V \setminus T') > \mu 2^i$. Note that $r \geq \operatorname{dist}(v, u) \geq \operatorname{dist}(T', B') \geq \operatorname{dist}(T', V \setminus T') > \mu 2^i$, and hence $\mu < r/2^i$. The decomposition $\mathcal{D}$ can therefore only cut $\beta_v(r)$ on level i if $\mu < r/2^i$. Since μ is chosen uniformly at random from the interval $(0, 1]$, the probability is less than $r/2^i$.

We now turn to the other case when $\beta_v(r)$ is cut by two small parts $A_1, A_2 \in \mathcal{A}_i^T$. The town T must have two child towns T_1 and T_2 for which $v \in T_1 \cap W \subseteq A_1$ and $u \in T_2 \cap W \subseteq A_2$. Let x_1

and x_2 be the connecting hubs of T_1 and T_2, respectively. The decomposition $\mathcal{D}$ cuts v and u on level i if and only if $\mathcal{X}_T$ cuts x_1 and x_2 on level $\ell = \min\{i, \lambda(X_T)\}$. Indeed, let U_1 and U_2 be the connecting i-clusters of T_1 and T_2, respectively, so that $A_1 = A^i_{U_1}$ and $A_2 = A^i_{U_2}$ with $x_1 \in U_1$ and $x_2 \in U_2$. Thus $\mathcal{D}$ cuts v and u on level i if and only if $U_1 \neq U_2$, i.e., if and only if $\mathcal{X}_T$ cuts x_1 and x_2 on level $\ell = \min\{i, \lambda(X_T)\}$.

To compute the probability that x_1 and x_2 are cut, it is necessary to bound the distance between them. As $v \in T_1$ and $u \in T_2$ while $u \in \beta_v(r)$, for each $j \in \{1, 2\}$ we have $\operatorname{dist}(T_j, V \setminus T_j) \leq \operatorname{dist}(T_1, T_2) \leq r$. By Lemma 12 the distance between T_j and its connecting hub $x_j \in X_T$ is thus at most $(1 + 2\rho)r$. Also, by Lemma 9 we have $\operatorname{diam}(T_j) < \operatorname{dist}(T_j, V \setminus T_j) \leq r$, and we get

$$\operatorname{dist}(x_1, x_2) \leq \operatorname{dist}(x_1, T_1) + \operatorname{diam}(T_1) + \operatorname{dist}(T_1, T_2) + \operatorname{diam}(T_2) + \operatorname{dist}(T_2, x_2) < 2(1 + 2\rho)r + 2r + r = (5 + 4\rho)r.$$

We can reformulate the above as follows: if $\mathcal{D}$ cuts the ball $\beta_v(r)$ at level i, and $\beta_v(r)$ is cut by some "small" parts A_1 and A_2, then $\mathcal{X}_T$ cuts the ball $\beta_{x_1}((5 + 4\rho)r)$ on level i, where x_1 is the hub defined for v above. We know that the probability of the latter event is at most $2^{O(d)}(5 + 4\rho)r/2^i$ by Lemma 4, where $d = O(\log(h \log(1/\rho)))$ is the doubling dimension of X_T by Theorem 8. Hence the probability that $\mathcal{D}$ cuts the ball $\beta_v(r)$ by some "small" parts at level i is at most $(h \log(1/\rho))^{O(1)} \cdot r/2^i$.

Taking a union bound over the two considered cases proves the claim. □

To prove the remaining property of Lemma 5 for $\mathcal{D}$, for each $B \in \mathcal{B}_i$ we need to choose an interface I_B from the whole vertex set V. For this we use a carefully chosen net (see Definition 6) of the hubs of the inclusion-wise minimal town T containing B, as formalized in the following lemma.

Lemma 16. *Given $B \in \mathcal{B}_i$ for some $\mathcal{B}_i \in \mathcal{D}$ where $i \geq 1$, let $T \in \mathcal{T}$ be the inclusion-wise minimal town containing B. We define the interface I_B to be a $\rho 2^i$-net of the set $Y_B = \{x \in X_T \mid \operatorname{dist}(x, B) \leq (1 + 2\rho) \operatorname{diam}(B)\}$. The interface I_B has the conciseness and preciseness properties of Lemma 5 for $\rho \leq 1/2$.*

Proof. We first prove that I_B is precise. Consider two vertices $u, v \in B$ that are cut at level $i - 1$ by $\mathcal{D}$. This means there are two distinct parts $B', B'' \in \mathcal{B}_{i-1}$ on this level such that $v \in B'$ and $u \in B''$. By definition, both B' and B'' are unions of sets $T' \cap W$ where T' is a child town of the inclusion-wise minimal town T containing B. Also $B' \cap B'' = \emptyset$ by Lemma 14. This means that T has two child towns T_1 and T_2 for which $v \in T_1 \cap W \subseteq B'$ and $u \in T_2 \cap W \subseteq B''$. By Theorem 8, there is a hub $x \in X_T$ such that $\operatorname{dist}(u, x) + \operatorname{dist}(x, v) \leq (1 + 2\rho) \operatorname{dist}(u, v)$. In particular, $\operatorname{dist}(x, B) \leq \operatorname{dist}(u, x) \leq (1 + 2\rho) \operatorname{dist}(u, v) \leq (1 + 2\rho) \operatorname{diam}(B)$, as $u, v \in B$. This means that $x \in Y_B$. Since I_B is a $\rho 2^i$-net of Y_B, there is a node $p \in I_B$ for which $\operatorname{dist}(x, p) \leq \rho 2^i$. By Lemma 13 we have $\operatorname{dist}(u, v) \leq \operatorname{diam}(B) \leq 2^{i+5}$ if $\rho \leq 1/2$, and so I_B is precise:

$$\begin{aligned} \operatorname{dist}(u, p) + \operatorname{dist}(p, v) &\leq \operatorname{dist}(u, x) + 2 \cdot \operatorname{dist}(x, p) + \operatorname{dist}(x, v) \\ &\leq (1 + 2\rho) \operatorname{dist}(u, v) + \rho 2^{i+1} \leq \operatorname{dist}(u, v) + 2\rho \cdot 2^{i+5} + \rho 2^{i+1} \leq \operatorname{dist}(u, v) + 68 \cdot \rho 2^i, \end{aligned}$$

To prove conciseness, recall that $\operatorname{diam}(B) \leq 2^{i+5}$ by Lemma 13, which means that $\operatorname{diam}(Y_B) \leq \operatorname{diam}(B) + 2(1 + 2\rho) \operatorname{diam}(B) \leq 5 \cdot 2^{i+5}$ for $\rho \leq 1/2$. Since I_B is a $\rho 2^i$-net of Y_B, Lemma 7 implies $|I_B| \leq 2^{d \cdot \lceil \log_2(160/\rho) \rceil}$, where d is the doubling dimension of Y_B. Theorem 8 says that X_T has doubling dimension $O(\log(h \log(1/\rho)))$, and as $Y_B \subseteq X_T$ the same asymptotic bound holds for the doubling dimension d of Y_B by Lemma 7. Therefore we get $|I_B| \leq 2^{O(\log(h \log(1/\rho)) \cdot (\log(1/\rho)))} \leq (h/\rho)^{O(1)}$, which concludes the proof. □

4 The algorithm

Let $\mathcal{I}$ be an instance of the k-Clustering^q or Facility Location^q problem on a shortest-path metric (V, dist) of a graph G with highway dimension h, and maximum demand $X = \max_{v \in V} \chi_{\mathcal{I}}(v)$. Given $\mathcal{I}$ the algorithm performs the following steps:

1. compute a town decomposition $\mathcal{T}$ of the metric together with the hub sets X_T for each town $T \in \mathcal{T}$ as given by Theorem 8.
2. compute a hierarchical decomposition $\mathcal{D}$ according to Lemma 5. Simultaneously $\mathcal{I}$ is reduced (see Section 4.1) to a *coarse* instance w.r.t. $\mathcal{D}$, meaning that there is a subset $W \subseteq V$ for which
   - the clients and facilities of $\mathcal{I}$ are contained in W, i.e., $F \cup \{v \in V \mid \chi_{\mathcal{I}}(v) > 0\} \subseteq W$, and
   - every part of $\mathcal{D}$ on level at most $\xi(W) = \lfloor \lambda(W) - 2\log_2(nX/\varepsilon) \rfloor$ has at most one facility, i.e., $|B \cap F| \leq 1$ for every $B \in \mathcal{B}_{\xi(W)}$.
3. compute the instance $\mathcal{I}_{\mathcal{D}}$ of small distortion as given by Lemma 11.
4. run a dynamic program on $\mathcal{I}_{\mathcal{D}}$ as given in Section 4.2, to compute an *optimum rounded interface-respecting solution* (see Section 4.1 for a formal definition), and convert it to a solution for the input instance.

In a nutshell, the coarseness of the instance guarantees that only a logarithmic number of levels need to be considered by the dynamic program. Reducing to a coarse instance in step (2) loses a $(1+\varepsilon)$-factor in the solution quality. The dynamic program is only able to compute highly structured solutions, which are captured by the notion of *rounding* and *interface-respecting*. Due to this, another $(1+\varepsilon)$-factor in the solution quality is lost. In Section 4.1 we prove that the output of the dynamic program corresponds to a near-optimal solution of the input instance (proving Theorem 2), and we also detail step (2) of the algorithm. Then in Section 4.2 we describe the dynamic program.

4.1 Approximating the distances

One caveat of the dynamic program is that the runtime is only polynomial if the the recursion depth is logarithmic. However when computing our decomposition on the whole metric (V, dist), the number of levels is $\lambda(V) + 1 = \lceil \log_2 \text{diam}(V) \rceil + 1$, which can be linear in the input size. For general metrics, standard preprocessing techniques can be used to reduce the number of levels to $O(\log(n/\varepsilon))$ when aiming for a $(1+\varepsilon)$-approximation. However, for graphs of bounded highway dimension these general techniques change the hub sets and we would have to be careful to maintain the properties we need in order to apply Theorem 8.[6] Therefore we adapt the standard techniques to our setting via the notion of *coarse* instances.

The following lemma shows that we can reduce any instance to a set of coarse ones, for which, as we will see, our dynamic program only needs to consider the highest $2\log_2(nX/\varepsilon)$ levels.

Lemma 17. *Let $\mathcal{I}$ be an instance of k-Clustering^q or Facility Location^q on a graph G of highway dimension h. There are polynomial-time computable instances $\mathcal{I}_1, \ldots, \mathcal{I}_b$ and respective hierarchical decompositions $\mathcal{D}_1, \ldots, \mathcal{D}_b$ with the properties given in Lemma 5 for any $\rho \leq 1/2$, such that for each $i \in \{1, \ldots, b\}$ the instance $\mathcal{I}_i$ is also defined on G and is coarse w.r.t. $\mathcal{D}_i$. Furthermore, if an α-approximation can be computed for each of the instances $\mathcal{I}_1, \ldots, \mathcal{I}_b$ in polynomial time, then for any $\varepsilon > 0$ a $(1+O(\varepsilon))\alpha$-approximation can be computed for $\mathcal{I}$ in polynomial time.*

Proof. Let us first describe the construction of the instances $\mathcal{I}_1, \ldots, \mathcal{I}_b$. We begin by computing a constant approximation L to the given instance $\mathcal{I}$ of k-Clustering^q or Facility Location^q, using

[6] We note that in [17] these general techniques are indeed applied to low highway dimension graphs, but some details of the argument are left out. Instead of rectifying the technique in [17], here we chose to go via the route of coarse instances.

a γ-approximation algorithm as given in [22] where $\gamma \in O(1)$. Let Λ be the value of the objective function of the approximate solution L, i.e., $\Lambda = \text{cost}_{\mathcal{I}}(L)$ if $\mathcal{I}$ is an instance of k-CLUSTERING^q and $\Lambda = \text{cost}_{\mathcal{I}}(L) + \sum_{f \in L} w_f$ in case of FACILITY LOCATION^q. Let Γ be the objective function value of an optimum solution OPT to $\mathcal{I}$. For every client c of $\mathcal{I}$ (for which $\chi_{\mathcal{I}}(c) > 0$), we have $\text{dist}(c, \text{OPT}) \le \Gamma^{1/q} \le \Lambda^{1/q}$. Hence if we consider the subgraph of G spanned by all edges of length at most $\Lambda^{1/q}$, then the closest facility of OPT to c lies in the same connected component of the subgraph as c.

Ideally, we would want each of these components to define an instance $\mathcal{I}_i$. However, such a component might not have bounded highway dimension and we would thus not be able to compute a hierarchical decomposition using Lemma 5 for it. Instead we use the same input graph $G = (V, E)$, but restrict the client and facility sets to a component. More formally, let $W_1, \dots, W_b \subseteq V$ be the vertex sets of the connected components of the subgraph of G spanned by all edges of length at most $\Lambda^{1/q}$. Note that $\text{diam}(W_i) \le n\Lambda^{1/q}$. For each $i \in \{1, \dots, b\}$ we define an instance $\mathcal{I}_i$ on G with $\chi_{\mathcal{I}_i}(v) = \chi_{\mathcal{I}}(v)$ for every $v \in W_i$ and $\chi_{\mathcal{I}_i}(v) = 0$ otherwise. Initially, the facility set F_i of $\mathcal{I}_i$ is $F \cap W_i$, where F is the facility set of $\mathcal{I}$. We still need to coarsen this set F_i though, which we do next.

At this point we compute a hierarchical decomposition $\mathcal{D}_i$ of W_i for each $\mathcal{I}_i$ using G according to Lemma 5, i.e., the interface sets are from $V \supseteq W_i$. To make $\mathcal{I}_i$ coarse w.r.t. $\mathcal{D}_i$, consider a part $B \in \mathcal{B}_{\xi(W_i)}$ of $\mathcal{D}_i$ containing facilities from F_i. In case of FACILITY LOCATION^q, let $f \in F_i \cap B$ be a facility of minimum weight w_f among those in $F_i \cap B$, and in case of k-CLUSTERING^q, fix an arbitrary $f \in F_i \cap B$. We call f the *representative facility of* $\mathcal{I}_i$ of the facilities in $F_i \cap B$, and remove all facilities other than f in $F_i \cap B$ from the set F_i. We repeat this for every part of $\mathcal{B}_{\xi(W)}$. Note that W_i contains all facilities and clients of $\mathcal{I}_i$, i.e., $F_i \cup \{v \in V \mid \chi_{\mathcal{I}_i}(v) > 0\} \subseteq W_i$, and thus $\mathcal{I}_i$ is now a coarse instance w.r.t. $\mathcal{D}_i$.

To prove the second part of the lemma, consider the optimum solution OPT to $\mathcal{I}$. For every i we define a solution S_i^* to $\mathcal{I}_i$, which for each facility in $\text{OPT} \cap W_i$ contains the representative facility of $\mathcal{I}_i$. Since in case of FACILITY LOCATION^q the representative facility is the one of minimum opening cost in the respective part in $\mathcal{B}_{\xi(W_i)}$ of $\mathcal{D}_i$ and the facility sets of different instances are disjoint, we have $\sum_{i=1}^{b} \sum_{f \in S_i^*} w_f \le \sum_{f \in \text{OPT}} w_f$. Also, $\sum_{i=1}^{b} |S_i^*| \le |\text{OPT}|$, which means that if $\mathcal{I}$ is an instance of k-CLUSTERING^q then each $\mathcal{I}_i$ should be an instance of k_i-CLUSTERING^q for $k_i = |S_i^*|$, where however we do not know the value of k_i a priori. We later show how to deal with this.

To bound the connection costs, first note that as $\lambda(W_i) = \lceil \log_2(\text{diam}(W_i)) \rceil$, $\text{diam}(W_i) \le n\Lambda^{1/q}$, and $1/q < 1 + 1/q \le 2$ for $q \ge 1$ we have

$$\xi(W_i) \le \lambda(W_i) - 2\log_2(nX/\varepsilon) < \log_2(n\Lambda^{1/q}) + 1 + \log_2\left(\frac{\varepsilon^{1+1/q}}{n^{1+1/q} X^{1/q}}\right) = \log_2\left(\varepsilon^{1+1/q} \left(\Lambda/(nX)\right)^{1/q}\right) + 1.$$

Now consider any client c of $\mathcal{I}_i$ and its closest facility $\hat{f} \in \text{OPT}$ in the optimum solution to $\mathcal{I}$, for which we know that $c, \hat{f} \in W_i$. Let $f^* \in S_i^*$ be the representative facility of $\hat{f}$, which lies in the same part $B \in \mathcal{B}_{\xi(W_i)}$ as $\hat{f}$. By Lemma 13 the diameter of B is less than $2^{\xi(W_i)+5}$ (if $\rho \le 1/2$). Hence we have

$$\text{dist}(c, f^*) \le \text{dist}(c, \hat{f}) + \text{diam}(B) < \text{dist}(c, \hat{f}) + 64 \cdot \left(\varepsilon^{1+1/q} \left(\Lambda/(nX)\right)^{1/q}\right).$$

To bound $\text{dist}(c, f^*)^q$ we need the following fact taken from [14].

Proposition 18 ([14])**.** *Given $x, y, q \ge 0$, and $0 < \varepsilon < 1/2$ we have $(x+y)^q \le (1+\varepsilon)^q x^q + (1+1/\varepsilon)^q y^q$.*

For constant $q \geq 1$ we have $(1+\varepsilon)^q = 1+O(\varepsilon)$ and $(1+1/\varepsilon)^q = O(1/\varepsilon^q)$ as ε tends to zero. Thus the bound of Proposition 18 can be stated as $(x+y)^q \leq (1+O(\varepsilon))x^q + O(1/\varepsilon^q)y^q$ if $q \geq 1$, and we get

$$\begin{aligned}\operatorname{dist}(c, f^*)^q &< (1+O(\varepsilon)) \operatorname{dist}(c, \hat{f})^q + O(1/\varepsilon^q)\left(\varepsilon^{1+1/q}\left(\Lambda/(nX)\right)^{1/q}\right)^q \\ &= (1+O(\varepsilon)) \operatorname{dist}(c, \hat{f})^q + O\left(\frac{\varepsilon\Lambda}{nX}\right).\end{aligned}$$

Using the definition of $\chi_{\mathcal{I}_i}(v)$, in addition to $\operatorname{dist}(c, S_i^*) \leq \operatorname{dist}(c, f^*)$, $\operatorname{dist}(c, \mathrm{OPT}) = \operatorname{dist}(c, \hat{f})$, and $\sum_{v \in V} \chi_{\mathcal{I}}(v) \leq nX$, we obtain

$$\begin{aligned}\sum_{i=1}^{b} \operatorname{cost}_{\mathcal{I}_i}(S_i^*) &= \sum_{i=1}^{b} \sum_{v \in W_i} \chi_{\mathcal{I}_i}(v) \cdot \operatorname{dist}(v, S_i^*)^q \\ &< \sum_{v \in V} \chi_{\mathcal{I}}(v)\left((1+O(\varepsilon)) \cdot \operatorname{dist}(v, \mathrm{OPT})^q + O\left(\frac{\varepsilon\Lambda}{nX}\right)\right) \\ &\leq (1+O(\varepsilon)) \operatorname{cost}_{\mathcal{I}}(\mathrm{OPT}) + O(\varepsilon\Lambda).\end{aligned}$$

For Facility Location^q, applying an α-approximation algorithm to each instance $\mathcal{I}_i$ gives respective solutions S_i for which

$$\begin{aligned}\sum_{i=1}^{b}\Big(\operatorname{cost}_{\mathcal{I}_i}(S_i) + \sum_{f \in S_i} w_f\Big) &\leq \sum_{i=1}^{b} \alpha\Big(\operatorname{cost}_{\mathcal{I}_i}(S_i^*) + \sum_{f \in S_i^*} w_f\Big) \\ &< \alpha\Big((1+O(\varepsilon)) \operatorname{cost}_{\mathcal{I}}(\mathrm{OPT}) + O(\varepsilon\Lambda) + \sum_{f \in \mathrm{OPT}} w_f\Big)\end{aligned}$$

As Λ is the objective function value of a γ-approximation to OPT where γ is constant, this means that by taking $\bigcup_{i=1}^{b} S_i$ as a solution to $\mathcal{I}$ we obtain a $(1+O(\varepsilon))\alpha$-approximation as required.

For k-Clustering^q we need to do more work, since we do not know the number of facilities k_i to be opened in each instance $\mathcal{I}_i$. First, for every $k' \in \{0, \ldots, k\}$ we compute an α-approximation $S_i(k')$ to k'-Clustering^q on each instance $\mathcal{I}_i$, i.e., $S_i(k') \subseteq F_i$ and $|S_i(k')| = k'$, and define $A_i(k') = \operatorname{cost}_{\mathcal{I}_i}(S_i(k'))$ to be its objective function value. Now let $A_{\leq i}(k')$ be of the minimum value of $\sum_{j=1}^{i} A_j(k'_j)$ over all tuples $k'_1, \ldots, k'_i$ for which $\sum_{j=1}^{i} k'_j = k'$. To compute $A_{\leq i}(k')$ in polynomial time, we use the following simple recursion. For $i = 1$ we clearly have $A_{\leq 1}(k') = A_1(k')$, and for $i > 1$ we have $A_{\leq i}(k') = \min\{A_{\leq i-1}(k' - k'_i) + A_i(k'_i) \mid 0 \leq k'_i \leq k'\}$. Note that it takes $O(bk^2)$ time to compute all values $A_{\leq i}(k')$. Finally, for the input instance $\mathcal{I}$ we output the union $\bigcup_{i=1}^{b} S_i(k'_i)$ of solutions that obtain the value $A_{\leq b}(k)$. By definition of $A_{\leq b}(k)$ this is a feasibly solution with k facilities, and we have $A_{\leq b}(k) \leq \sum_{i=1}^{b} A_i(k_i)$ for the values $k_i = |S_i^*|$. Thus

$$\begin{aligned}\sum_{i=1}^{b} \operatorname{cost}_{\mathcal{I}_i}(S_i(k'_i)) = A_{\leq b}(k) \leq \sum_{i=1}^{b} \operatorname{cost}_{\mathcal{I}_i}(S_i(k_i)) &\leq \sum_{i=1}^{b} \alpha \operatorname{cost}_{\mathcal{I}_i}(S_i^*) \\ &\leq \alpha\Big((1+O(\varepsilon)) \operatorname{cost}_{\mathcal{I}}(\mathrm{OPT}) + O(\varepsilon\Lambda)\Big).\end{aligned}$$

Hence the output $\bigcup_{i=1}^{b} S_i(k'_i)$ is a $(1+O(\varepsilon))\alpha$-approximation, since Λ is a constant approximation of OPT. □

Lemma 17 implies that if there is a PTAS for coarse instances, we also have a PTAS in general. Hence from now on we assume that the given instance $\mathcal{I}$ is coarse w.r.t. a hierarchical decomposition $\mathcal{D}$ of some subset W of the vertices of the input graph G, where $\mathcal{D}$ has bounded scaling probability factor and concise and precise interface sets in G according to Lemma 5 (for some value $\rho > 0$ specified later)

The next step of the algorithm is to compute a new instance $\mathcal{I}_{\mathcal{D}}$ with small distortion as given by Lemma 11. Recall that $\mathcal{I}_{\mathcal{D}}$ is obtained from $\mathcal{I}$ by moving badly cut clients to facilities of L. In particular, the instance $\mathcal{I}_{\mathcal{D}}$ is also coarse w.r.t. $\mathcal{D}$, which means that we may run our dynamic program on $\mathcal{I}_{\mathcal{D}}$.

The dynamic program exploits the interface sets of $\mathcal{D}$ by computing a near-optimum "interface-respecting" solution to $\mathcal{I}_{\mathcal{D}}$, i.e., a solution where clients are connected to facilities through interface points. Moreover, for the dynamic program to run in polynomial time it can only estimate the distances between interface points and facilities to a certain precision. In general, we denote by $\langle x\rangle_i = \min\{(69+\delta)\rho 2^i \mid \delta \in \mathbb{N} \text{ and } \rho\delta 2^i \geq x\}$ the value of x rounded to the next multiple of $\rho 2^i$ and shifted by $69\rho 2^i$. We then define the *rounded interface-respecting distance* $\mathrm{dist}'(v,u)$ from a vertex v to another vertex u as follows. If $v = u$ then $\mathrm{dist}'(v,u) = 0$. Otherwise, let $i \geq 1$ be the level of $\mathcal{D}$ such that there is a part $B \in \mathcal{B}_i$ with $v, u \in B$, and $\mathcal{D}$ cuts v and u at level $i-1$. We let

$$\mathrm{dist}'(v,u) = \min\left\{ \mathrm{dist}(v,p) + \langle \mathrm{dist}(p,u)\rangle_i \mid p \in I_B \right\}.$$

Note that $\mathrm{dist}'(\cdot,\cdot)$ does not necessarily fulfill the triangle inequality, and is also not symmetric. We therefore need the bounds of the following lemma.

Lemma 19. *For any level $i \geq 1$ and vertices v and u that are cut by $\mathcal{D}$ on level $i-1$ we have $\mathrm{dist}'(v,u) \leq \mathrm{dist}(v,u) + 138\cdot\rho 2^i$. Let $B \in \mathcal{B}_j$ be the part on some level $j \geq i$ with $v, u \in B$. For any $p \in I_B$ we have $\mathrm{dist}'(v,u) \leq \mathrm{dist}(v,u) + \langle \mathrm{dist}(p,u)\rangle_j$.*

Proof. Let $B' \in \mathcal{B}_i$ be the part on level i containing both v and u. By Lemma 5 there is an interface point $p' \in I_{B'}$ such that $\mathrm{dist}(v,p') + \mathrm{dist}(p',u) \leq \mathrm{dist}(v,u) + 68\cdot\rho 2^i$. By definition of the rounding we also have $\langle \mathrm{dist}(p',u)\rangle_i \leq \mathrm{dist}(p',u) + 70\cdot\rho 2^i$. Hence $\mathrm{dist}'(v,u) \leq \mathrm{dist}(v,p') + \langle\mathrm{dist}(p',u)\rangle_i \leq \mathrm{dist}(v,p') + \mathrm{dist}(p',u) + 70\cdot\rho 2^i \leq \mathrm{dist}(v,u) + 138\cdot\rho 2^i$.

The second part is obvious if $j = i$ from the definition of $\mathrm{dist}'(v,u)$. If $j \geq i+1$, we use the above bound on $\mathrm{dist}'(v,u)$ together with the additive shift of the rounding and the triangle inequality of $\mathrm{dist}(\cdot,\cdot)$ to obtain

$$\begin{aligned}\mathrm{dist}'(v,u) &\leq \mathrm{dist}(v,u) + 138\cdot\rho 2^i \leq \mathrm{dist}(v,p) + \mathrm{dist}(p,u) + 138\cdot\rho 2^{j-1} \\ &\leq \mathrm{dist}(v,p) + \langle\mathrm{dist}(p,u)\rangle_j - 69\cdot\rho 2^j + 138\cdot\rho 2^{j-1} = \mathrm{dist}(v,p) + \langle\mathrm{dist}(p,u)\rangle_j. \quad \square\end{aligned}$$

For any non-empty set S of facilities, we define $\mathrm{dist}'(v,S) = \min_{f\in S}\{\mathrm{dist}'(v,S)\}$, and for empty sets we let $\mathrm{dist}'(v,\emptyset) = \infty$. Analogous to $\mathrm{cost}_{\mathcal{I}_0}(S)$, for a solution S to some instance $\mathcal{I}_0$ we also define $\mathrm{cost}'_{\mathcal{I}_0}(S)$ using $\mathrm{dist}'(\cdot,\cdot)$ as

$$\mathrm{cost}'_{\mathcal{I}_0}(S) = \sum_{v\in V} \chi_{\mathcal{I}_{\mathcal{D}}}(v)\cdot \mathrm{dist}'(v,S)^q.$$

We show the following lemma, which translates between $\mathrm{cost}'_{\mathcal{I}_{\mathcal{D}}}$ and $\mathrm{cost}_{\mathcal{I}}$, and is implied by the preciseness of the interface sets and the fact that $\mathcal{I}_{\mathcal{D}}$ has small distortion. Recall that this means that $\nu_{\mathcal{I}_{\mathcal{D}}} \leq \varepsilon\,\mathrm{cost}_{\mathcal{I}}(L)$, and there exists a witness solution $\hat{S} \subseteq F$ that contains the badly cut facilities $B_{\mathcal{D}}$ and for which $\mathrm{cost}_{\mathcal{I}_{\mathcal{D}}}(\hat{S}) \leq (1+O(\varepsilon))\,\mathrm{cost}_{\mathcal{I}}(\mathrm{OPT}) + O(\varepsilon)\,\mathrm{cost}_{\mathcal{I}}(L)$. Moreover, in the case of Facility Location^q, $\hat{S} = \mathrm{OPT} \cup B_{\mathcal{D}}$ and $\sum_{f\in B_{\mathcal{D}}} w_f \leq \varepsilon\cdot\sum_{f\in L} w_f$. Recall also that the set of facilities is the same in $\mathcal{I}$ and $\mathcal{I}_{\mathcal{D}}$, i.e., a solution to one of these instances is also a solution to the other.

Lemma 20. *Let $\mathcal{I}$ be an instance of k-*Clustering^q *or* Facility Location^q *with optimum solution* OPT *and approximate solution L. Let $\mathcal{I}_\mathcal{D}$ be an instance of small distortion for some $0 < \varepsilon < 1/2$, computed from L and a hierarchical decomposition $\mathcal{D}$ with precise interface sets for $\rho \le \frac{\varepsilon^{q+4+1/q}}{1104\cdot\sigma(q+1)^q}$ according to Lemma 5. For the witness solution $\hat{S}$ of $\mathcal{I}_\mathcal{D}$ we have $\mathrm{cost}'_{\mathcal{I}_\mathcal{D}}(\hat{S}) \le (1+O(\varepsilon))\,\mathrm{cost}_\mathcal{I}(\mathrm{OPT}) + O(\varepsilon)\,\mathrm{cost}_\mathcal{I}(L)$. Moreover, for any solution S we have $\mathrm{cost}_\mathcal{I}(S) \le (1+O(\varepsilon))\,\mathrm{cost}'_{\mathcal{I}_\mathcal{D}}(S) + O(\varepsilon)\,\mathrm{cost}_\mathcal{I}(L)$.*

Proof. To show the first inequality, we consider the rounded connection costs of clients to their closest facility in $\hat{S}$ via some interface point. That is, let c be a client of $\mathcal{I}_\mathcal{D}$ and let $f \in \hat{S}$ be its closest facility (according to $\mathrm{dist}(\cdot,\cdot)$). If $c \ne f$, there is a level $i \ge 1$ for which $\mathcal{D}$ cuts c and f at level $i-1$. By Lemma 19 we have $\mathrm{dist}'(c,f) \le \mathrm{dist}(c,f) + 138\cdot\rho 2^i$. Also, by Lemma 11 we know that $i - 1 \le \log_2(3L_{\tilde{c}}/\varepsilon + 4\mathrm{OPT}_{\tilde{c}}) + \tau(\varepsilon,q,\sigma)$, where $L_{\tilde{c}}$ and $\mathrm{OPT}_{\tilde{c}}$ are the respective minimum distances from the original position $\tilde{c}$ of c to L and OPT in $\mathcal{I}$. Hence using the definitions of $\tau(\varepsilon,q,\sigma) = \log_2(\sigma(q+1)^q/\varepsilon^{q+1})$ and $\rho \le \frac{\varepsilon^{q+4+1/q}}{1104\cdot\sigma(q+1)^q}$ we get

$$
\begin{aligned}
\mathrm{dist}'(c,f) &\le \mathrm{dist}(c,f) + 138\cdot\rho 2^i \\
&\le \mathrm{dist}(c,f) + 138\cdot\rho 2^{\log_2(3L_{\tilde{c}}/\varepsilon+4\mathrm{OPT}_{\tilde{c}})+\tau(\varepsilon,q,\sigma)+1} \\
&\le \mathrm{dist}(c,f) + 138\cdot 4(L_{\tilde{c}} + \mathrm{OPT}_{\tilde{c}})\cdot 2^{\tau(\varepsilon,q,\sigma)+1}/\varepsilon \\
&\le \mathrm{dist}(c,f) + (L_{\tilde{c}} + \mathrm{OPT}_{\tilde{c}})\cdot 1104\cdot\rho\sigma(q+1)^q/\varepsilon^{q+2} \\
&\le \mathrm{dist}(c,f) + \varepsilon^{2+1/q}(L_{\tilde{c}} + \mathrm{OPT}_{\tilde{c}}).
\end{aligned}
$$

If $c = f$ we have $\mathrm{dist}'(c,f) = 0 = \mathrm{dist}(c,f)$, and so the above inequality holds trivially.

To bound $\mathrm{dist}'(c,f)^q$, we use the bound of Proposition 18, which can be stated as $(x+y)^q \le (1+O(\varepsilon))x^q + O(1/\varepsilon^q)y^q$ if $q \ge 1$. Applying this twice to the bound on $\mathrm{dist}'(c,f)$ above, we get

$$
\begin{aligned}
\mathrm{dist}'(c,f)^q &\le (1+O(\varepsilon))\,\mathrm{dist}(c,f)^q + O(\varepsilon^{q+1})(L_{\tilde{c}} + \mathrm{OPT}_{\tilde{c}})^q \\
&\le (1+O(\varepsilon))\,\mathrm{dist}(c,f)^q + O(\varepsilon^{q+1}(1+\varepsilon))L^q_{\tilde{c}} + O(\varepsilon)\mathrm{OPT}^q_{\tilde{c}} \\
&\le (1+O(\varepsilon))\,\mathrm{dist}(c,f)^q + O(\varepsilon)(L^q_{\tilde{c}} + \mathrm{OPT}^q_{\tilde{c}}).
\end{aligned}
$$

To bound $\mathrm{cost}'_{\mathcal{I}_\mathcal{D}}(\hat{S})$ using this inequality we define $L_{\tilde{v}} = \mathrm{OPT}_{\tilde{v}} = 0$ for any non-client v of $\mathcal{I}_\mathcal{D}$, i.e., whenever $\chi_{\mathcal{I}_\mathcal{D}}(v) = 0$, so that applying the definition of $\chi_{\mathcal{I}_\mathcal{D}}$ we obtain

$$
\begin{aligned}
\mathrm{cost}'_{\mathcal{I}_\mathcal{D}}(\hat{S}) &= \sum_{v\in V} \chi_{\mathcal{I}_\mathcal{D}}(v)\cdot\mathrm{dist}'(v,\hat{S})^q \\
&\le \sum_{v\in V} \chi_{\mathcal{I}_\mathcal{D}}(v)\left((1+O(\varepsilon))\cdot\mathrm{dist}(v,\hat{S})^q + O(\varepsilon)(L^q_{\tilde{v}} + \mathrm{OPT}^q_{\tilde{v}})\right) \\
&= (1+O(\varepsilon))\,\mathrm{cost}_{\mathcal{I}_\mathcal{D}}(\hat{S}) + O(\varepsilon)(\mathrm{cost}_\mathcal{I}(L) + \mathrm{cost}_\mathcal{I}(\mathrm{OPT})).
\end{aligned}
$$

Since $\hat{S}$ is the witness solution of $\mathcal{I}_\mathcal{D}$, we know that $\mathrm{cost}_{\mathcal{I}_\mathcal{D}}(\hat{S}) \le (1+O(\varepsilon))\,\mathrm{cost}_\mathcal{I}(\mathrm{OPT}) + O(\varepsilon)\,\mathrm{cost}_\mathcal{I}(L)$ so that also $\mathrm{cost}'_{\mathcal{I}_\mathcal{D}}(\hat{S}) \le (1+O(\varepsilon))\,\mathrm{cost}_\mathcal{I}(\mathrm{OPT}) + O(\varepsilon)\,\mathrm{cost}_\mathcal{I}(L)$, as claimed.

For the second inequality of the lemma for any solution S, since $\mathcal{I}_\mathcal{D}$ has small distortion we have $\mathrm{cost}_\mathcal{I}(S) - (1+2\varepsilon)\,\mathrm{cost}_{\mathcal{I}_\mathcal{D}}(S) \le \nu_{\mathcal{I}_\mathcal{D}} \le \varepsilon\,\mathrm{cost}_\mathcal{I}(L)$. This immediately implies $\mathrm{cost}_\mathcal{I}(S) \le (1+2\varepsilon)\,\mathrm{cost}'_{\mathcal{I}_\mathcal{D}}(S) + \varepsilon\,\mathrm{cost}_\mathcal{I}(L)$, since $\mathrm{dist}(c,f) \le \mathrm{dist}'(c,f)$ by the triangle inequality of $\mathrm{dist}(\cdot,\cdot)$ and the fact that $\langle x\rangle_i \ge x$ for any x. □

The next lemma states the properties of the dynamic program that for any coarse instance $\mathcal{I}_0$ computes an *optimal rounded interface-respecting solution*, which formally is a subset OPT' of facilities that minimizes $\mathrm{cost}'_{\mathcal{I}_0}(\mathrm{OPT}')$ with $|\mathrm{OPT}'| \leq k$ for k-CLUSTERING^q, while for FACILITY LOCATION^q it minimizes $\mathrm{cost}'_{\mathcal{I}_0}(\mathrm{OPT}') + \sum_{f \in \mathrm{OPT}'} w_f$. This step of the algorithm exploits the conciseness of the interface sets and the coarseness of the instance to bound the runtime. We prove the following lemma in Section 4.2.

Lemma 21. *Let $\mathcal{I}_0$ be an instance of k-CLUSTERING^q or FACILITY LOCATION^q that for some $\varepsilon > 0$ is coarse w.r.t. a hierarchical decomposition $\mathcal{D}$ with concise interface sets for some $1/2 \geq \rho > 0$ according to Lemma 5. An optimum rounded interface-respecting solution for $\mathcal{I}_0$ can be computed in $(nX/\varepsilon)^{(h/\rho)^{O(1)}}$ time.*

We are now ready to put together the above lemmas to prove Theorem 2.

Proof Theorem 2. Given an instance of k-CLUSTERING^q or FACILITY LOCATION^q we first apply Lemma 17 to reduce to a coarse instance. Lemma 17 also supplies a hierarchical decomposition $\mathcal{D}$ with the properties given in Lemma 5. We use this together with a constant approximation L of the coarse instance $\mathcal{I}$ to compute a new instance $\mathcal{I}_\mathcal{D}$ with small distortion via Lemma 11. On this instance we apply Lemma 21 to compute an optimum rounded interface-respecting solution OPT' in $(nX/\varepsilon)^{(h/\rho)^{O(1)}}$ time. Since the facility sets of $\mathcal{I}$ and $\mathcal{I}_\mathcal{D}$ are the same, we may output OPT' for $\mathcal{I}$, which can then be converted into a solution of the original non-coarse input instance using Lemma 17, while only losing a $(1+O(\varepsilon))$-factor in the objective function. Hence it suffices to show that OPT' is a $(1+O(\varepsilon))$-approximation to $\mathcal{I}$ and to bound the runtime of the algorithm.

From Lemma 20 we get $\mathrm{cost}_\mathcal{I}(\mathrm{OPT}') \leq (1+O(\varepsilon))\,\mathrm{cost}'_{\mathcal{I}_\mathcal{D}}(\mathrm{OPT}') + O(\varepsilon)\,\mathrm{cost}_\mathcal{I}(L)$ by setting $\rho \leq \frac{\varepsilon^{q+4+1/q}}{1104 \cdot \sigma (q+1)^q}$. We know that $\mathrm{cost}'_{\mathcal{I}_\mathcal{D}}(\mathrm{OPT}') \leq \mathrm{cost}'_{\mathcal{I}_\mathcal{D}}(\hat{S})$ for k-CLUSTERING^q, where $\hat{S}$ is the witness solution of $\mathcal{I}_\mathcal{D}$. Putting these inequalities together we have $\mathrm{cost}_\mathcal{I}(\mathrm{OPT}') \leq \mathrm{cost}'_{\mathcal{I}_\mathcal{D}}(\hat{S}) + O(\varepsilon)\,\mathrm{cost}_\mathcal{I}(\mathrm{OPT})$, since L is a constant approximation to the optimum solution OPT to $\mathcal{I}$. For the same reason, Lemma 20 also implies that $\mathrm{cost}'_{\mathcal{I}_\mathcal{D}}(\hat{S}) \leq (1+O(\varepsilon))\,\mathrm{cost}_\mathcal{I}(\mathrm{OPT})$, which gives $\mathrm{cost}_\mathcal{I}(\mathrm{OPT}') \leq (1+O(\varepsilon))\,\mathrm{cost}_\mathcal{I}(\mathrm{OPT})$, i.e., for k-CLUSTERING^q the solution OPT' is a $(1+O(\varepsilon))$-approximation to OPT.

For FACILITY LOCATION^q we have $\mathrm{cost}'_{\mathcal{I}_\mathcal{D}}(\mathrm{OPT}') + \sum_{f \in \mathrm{OPT}'} w_f \leq \mathrm{cost}'_{\mathcal{I}_\mathcal{D}}(\hat{S}) + \sum_{f \in \hat{S}} w_f$, which by the above bounds gives $\mathrm{cost}_\mathcal{I}(\mathrm{OPT}') + \sum_{f \in \mathrm{OPT}'} w_f \leq (1+O(\varepsilon))\,\mathrm{cost}_\mathcal{I}(\mathrm{OPT}) + \sum_{f \in \hat{S}} w_f$. In case of FACILITY LOCATION^q we have the additional property that $\hat{S}$ is the union of OPT and the badly cut clients $B_\mathcal{D}$, which implies $\sum_{f \in \hat{S}} w_f \leq \sum_{f \in \mathrm{OPT}} w_f + \sum_{f \in B_\mathcal{D}} w_f$. Furthermore, we have $\sum_{f \in B_\mathcal{D}} w_f \leq \varepsilon \cdot \sum_{f \in L} w_f$, and hence $\mathrm{cost}_\mathcal{I}(\mathrm{OPT}') + \sum_{f \in \mathrm{OPT}} w_f \leq (1+O(\varepsilon))\,\mathrm{cost}_\mathcal{I}(\mathrm{OPT}) + \sum_{f \in \mathrm{OPT}} w_f + \varepsilon \cdot \sum_{f \in L} w_f$. Again using that L is a constant approximation of OPT we obtain $\mathrm{cost}_\mathcal{I}(\mathrm{OPT}') + \sum_{f \in \mathrm{OPT}} w_f \leq (1+O(\varepsilon))(\mathrm{cost}_\mathcal{I}(\mathrm{OPT}) + \sum_{f \in \mathrm{OPT}} w_f)$, i.e., also in this case OPT' is a $(1+O(\varepsilon))$-approximation to OPT.

Next we bound the runtime. According to Lemma 20 we need to set $\rho \leq \frac{\varepsilon^{q+4+1/q}}{1104 \cdot \sigma (q+1)^q}$, while according the scaling probability factor we obtain from Lemma 5 is $\sigma \leq (h \log(1/\rho))^c$ for some constant c. Note that the bound on ρ depends on σ and vice versa, which means that we need to be careful when determining a value for ρ respecting the bound from Lemma 20. In particular, substituting the bound for σ in the bound for ρ and rearranging, it suffices to set ρ such that $\rho \log^c(1/\rho) \leq \frac{\varepsilon^{q+4+1/q}}{1104 \cdot (q+1)^q h^c}$. Observe that for any value $x > 0$ for which $\log^c(1/x^2) \leq 1/x$, setting $\rho = \frac{x}{\log^c(1/x^2)}$ we have $\rho \log^c(1/\rho) = x \cdot \frac{\log^c(\log^c(1/x^2) \cdot \frac{1}{x})}{\log^c(1/x^2)} \leq x$. Since there exists some constant c' such that $\log^c(1/x^2) \leq 1/x$ for any $x \in (0, c']$, for sufficiently small ε we can set $x = \frac{\varepsilon^{q+4+1/q}}{1104 \cdot (q+1)^q h^c}$ so that the inequality of Lemma 20 is fulfilled (note that ε can be chosen independent of h and q). Setting x this way also implies $\rho = \frac{x}{\log^c(1/x^2)} \geq x^2 \geq (\frac{\varepsilon}{hq})^{\Theta(q)}$, and thus according to Lemma 21 the runtime of the dynamic program becomes

$$(nX/\varepsilon)^{(h/\rho)^{O(1)}} \leq (nX)^{(hq/\varepsilon)^{O(q)}}.$$

All other steps of the algorithm run in polynomial time, and so the claimed runtime follows. □

4.2 The dynamic program (proof of Lemma 21)

We describe the algorithm for k-Clustering^q, and only mention in the end how to modify the algorithm to compute a solution for Facility Location^q. We develop a dynamic program, which follows more or less standard techniques as for instance given in [5, 24], but needs to handle the weaker preciseness property of the interface.

The solution is computed by a dynamic program recursing on the decomposition $\mathcal{D}$. Let W be the vertex set that $\mathcal{D}$ decomposes, and which contains all clients and facilities of the coarse instance $\mathcal{I}$. Roughly speaking, the table of the dynamic program will have an entry for every part $B \in \mathcal{B}_i$ of $\mathcal{D}$ on all levels $i \geq \xi(W)$, for which it will estimate the distance from each interface point on all higher levels $j \geq i+1$ to the closest facility of the optimum solution. That is, if $\tilde{B} \in \mathcal{B}_j$ is a higher-level part for which $B \subseteq \tilde{B}$, then the distances from all interface points $I_{\tilde{B}}$ to facilities of the solution in $\tilde{B}$ will be estimated for B.

Here the estimation happens in two ways. First off, the distances to facilities outside of B have to be guessed. That is, there is an *external* distance function d_j^+ that assigns a distance to each interface point of $I_{\tilde{B}}$, anticipating the distance from such a point to the closest facility of $\tilde{B}$, if this facility lies outside of B. In order to verify whether the guess was correct, each entry for a part B on level i also provides an *internal* distance function d_j^-, which stores the distance from each interface point of $I_{\tilde{B}}$ on level $j \geq i+1$ to the closest facility, if the facility is guessed to lie inside of B.

The other way in which distances are estimated concerns the preciseness with which they are stored. The distance functions d_j^+ and d_j^- will only take rounded values $\langle x \rangle_j$ where $0 < x \leq 2^{j+6}$, or ∞ if no facility at the appropriate distance exists. In particular, if the facility of the solution in $\tilde{B}$ that is closest to $p \in I_{\tilde{B}}$ lies outside of B then $d_j^-(p) = \infty$, and if it lies inside of B then $d_j^+(p) = \infty$. If there is no facility of the solution in $\tilde{B}$ then both distance functions d_j^+ and d_j^- are set to ∞ for all $p \in I_{\tilde{B}}$. Note that this means that at least one of $d_j^+(p)$ and $d_j^-(p)$ is always set to ∞. Note also that the finite values in the domains of the distance functions admit to store the rounded distance to any facility in $\tilde{B}$ on level j, since the diameter of $\tilde{B}$ is at most 2^{j+5} by Lemma 13, and the distance from any $p \in I_{\tilde{B}}$ to $\tilde{B}$ is at most $(1+2\rho)\operatorname{diam}(\tilde{B})$ by Lemma 16, i.e., for any $f \in \tilde{B} \cap F$ we have $\operatorname{dist}(p, f) \leq (1+2\rho)2^{j+5} \leq 2^{j+6}$ using $\rho \leq 1/2$.

Formal definition of the table. Let us denote by I_B^j the interface set of the part $\tilde{B} \in \mathcal{B}_j$ on level $j \geq i+1$ containing $B \in \mathcal{B}_i$, i.e., $I_B^j = I_{\tilde{B}}$. Every entry of the dynamic programming table T is defined by a part $B \in \mathcal{B}_i$ of $\mathcal{D}$ on a level $i \in \{\xi(W), \dots, \lambda(W)\}$, and two distance functions $d_j^+, d_j^- : I_B^j \to \{\langle x \rangle_j \mid 0 < x \leq 2^{j+6}\} \cup \{\infty\}$ for each $j \in \{i+1, \dots, \lambda(W)\}$, such that $\max\{d_j^+(p), d_j^-(p)\} = \infty$ for all $p \in I_B^j$. Additionally, each entry comes with an integer $k' \in \{0, \dots, k\}$, which is a guess on the number of facilities that the optimum solution contains in B.

In an entry $T[B, k', (d_j^+, d_j^-)_{j=i+1}^{\lambda(W)}]$ we store the rounded interface-respecting cost of connecting the clients of B to facilities that adhere to the distance functions. More concretely, let $S \subseteq F \cap B$ be any subset of facilities in B. We say that S is *compatible* with an entry $T[B, k', (d_j^+, d_j^-)_{j=i+1}^{\lambda(W)}]$ if $|S| = k'$, and for any $j \geq i+1$ the values of the distance functions for every interface point $p \in I_B^j$ are set to either

- $d_j^-(p) = \langle \mathrm{dist}(p,S)\rangle_j$ and $d_j^+(p) = \infty$, or
- $d_j^+(p) \leq \langle \mathrm{dist}(p,S)\rangle_j$ and $d_j^-(p) = \infty$.

Recall that $\mathrm{dist}(v,\emptyset) = \infty$, and so the empty set $S = \emptyset$ is compatible with an entry $T[B,k',(d_j^+,d_j^-)_{j=i+1}^{\lambda(W)}]$ if $k' = 0$, and the values of all internal distance functions are set to ∞. An entry $T[B,k',(d_j^+,d_j^-)_{j=i+1}^{\lambda(W)}]$ for $B \in \mathcal{B}_i$ should store the minimum value $C_B(S)$ over all sets $S \subseteq F \cap B$ compatible with the entry, where $C_B(S)$ is defined as

$$C_B(S) = \sum_{v\in B} \chi_{\mathcal{I}_0}(v) \cdot \min\Big\{ \mathrm{dist}'(v,S), \min_{\substack{j\geq i+1 \\ p\in I_B^j}} \big\{ \mathrm{dist}(v,p) + d_j^+(p) \big\}\Big\}.$$

If there is no compatible set $S \subseteq F \cap B$ for the entry, then $T[B,k',(d_j^+,d_j^-)_{j=i+1}^{\lambda(W)}] = \infty$.

On the highest level $i = \lambda(W)$, there are no distance functions to adhere to on levels $j \geq i+1$, and thus any set $S \subseteq W$ of facilities is compatible with the entry for $B = W$ and $k' = |S|$. Furthermore, $\mathrm{cost}'_{\mathcal{I}_0}(S)$ is equal to $C_W(S)$, since W contains all clients and facilities of the coarse instance $\mathcal{I}_0$. In particular, the entry of T for which $k' = k$ and $B = W$, will contain the objective function value of the optimum rounded interface-respecting solution to $\mathcal{I}_0$. Hence if we can compute the table T we can also output the optimum rounded interface-respecting solution via this entry.

Computing the table. We begin with a part $B \in \mathcal{B}_{\xi(W)}$ on the lowest considered level $\xi(W)$, for which we know that B contains at most one facility, as $\mathcal{I}_0$ is coarse. If B contains no facility, then only $S = \emptyset$ can be compatible with the entry $T[B,k',(d_j^+,d_j^-)_{j=i+1}^{\lambda(W)}]$ and computing the value of the entry is straightforward given the definition of $C_B(S)$, where all incompatible entries are set to ∞. If B contains one facility f, then any compatible set S is either empty or only contains f. We can thus check whether either of the two options is compatible with the entry $T[B,k',(d_j^+,d_j^-)_{j=i+1}^{\lambda(W)}]$ by checking if k' is set to 0 or 1, respectively, and checking that all values of the internal distance function are set correctly. Thereafter we can again use the definition of $C_B(S)$ to compute the values for both possible sets S and store them in the respective compatible entries. All incompatible entries are set to ∞.

Now fix a part $B \in \mathcal{B}_i$ that lies on a level $i > \xi(W)$. We show how to compute all entries $T[B,k',(d_j^+,d_j^-)_{j=i+1}^{\lambda(W)}]$ for all values k' and distance functions recursively. By induction we have already computed the correct values of all entries of T for parts $B' \in \mathcal{B}_{i-1}$ where $B' \subseteq B$. We order these parts arbitrarily, so that $B'_1,\dots,B'_b$ are the parts of $\mathcal{B}_{i-1}$ contained in B. We then define an auxiliary table $\hat{T}$ that is similar to the table T, but should compute the best compatible facility set in the union $B'_{\leq\ell} = \bigcup_{h=1}^{\ell} B'_h$ of the first ℓ subparts of B. Accordingly, $\hat{T}$ has an entry for each union of parts $B'_{\leq\ell}$, each $k' \in \{0,\dots,k\}$, and distance functions $d_j^+, d_j^- : I_B^j \to \{\langle x\rangle_j \mid 0 < x \leq 2^{j+6}\} \cup \{\infty\}$ for each $j \in \{i,\dots,\lambda(W)\}$, such that $\max\{d_j^+(p), d_j^-(p)\} = \infty$ for all $p \in I_B^j$. Here, naturally, $I_B^i = I_B$, i.e., the entry also takes the interface set of B into account.

Analogous to before, a set $S \subseteq F \cap B'_{\leq\ell}$ of facilities in the union is *compatible* with an entry $\hat{T}[B'_{\leq\ell},k',(d_j^+,d_j^-)_{j=i}^{\lambda(W)}]$ if $|S| = k'$, and for any $j \geq i$ the values of the distance functions for every interface point $p \in I_B^j$ are set to either

- $d_j^-(p) = \langle \mathrm{dist}(p,S)\rangle_j$ and $d_j^+(p) = \infty$, or
- $d_j^+(p) \leq \langle \mathrm{dist}(p,S)\rangle_j$ and $d_j^-(p) = \infty$.

The entry $\hat{T}[B'_{\leq\ell},k',(d_j^+,d_j^-)_{j=i}^{\lambda(W)}]$ should store the minimum value of $\hat{C}_{\leq\ell}(S)$ over all compatible sets $S \subseteq F \cap B'_{\leq \ell}$, where $\hat{C}_{\leq \ell}(S)$ is defined as

$$\hat{C}_{\leq \ell}(S) = \sum_{v \in B'_{\leq \ell}} \chi_{\mathcal{I}_0}(v) \cdot \min \Big\{ \operatorname{dist}'(v, S), \min_{\substack{j \geq i \\ p \in I_B^j}} \big\{ \operatorname{dist}(v, p) + d_j^+(p) \big\} \Big\}.$$

If there is no compatible set $S \subseteq F \cap B'_{\leq \ell}$ for the entry, then $\hat{T}[B'_{\leq \ell}, k', (d_j^+, d_j^-)_{j=i}^{\lambda(W)}] = \infty$.

To compute T using the auxiliary table $\hat{T}$, note that since $B = B'_{\leq b}$, any set $S \subseteq F \cap B$ is compatible with the entry $T[B, k', (d_j^+, d_j^-)_{j=i+1}^{\lambda(W)}]$ if and only if it is compatible with a corresponding entry $\hat{T}[B'_{\leq b}, k', (d_j^+, d_j^-)_{j=i}^{\lambda(W)}]$ for some internal distance function d_i^- on level i. Furthermore, if $d_i^+(p) = \infty$ for all $p \in I^i$, then $C_B(S) = \hat{C}_{\leq b}(S)$ for such a set S. Therefore we can easily compute the entry $T[B, k', (d_j^+, d_j^-)_{j=i+1}^{\lambda(W)}]$ from $\hat{T}$ by setting

$$T[B, k', (d_j^+, d_j^-)_{j=i+1}^{\lambda(W)}] = \min_{d_i^-} \Big\{ \hat{T}[B'_{\leq b}, k', (d_j^+, d_j^-)_{j=i}^{\lambda(W)}] \mid \forall p \in I_B^i : d_i^+(p) = \infty \Big\}.$$

Computing the auxiliary table. Also computing an entry of $\hat{T}$ for $B'_{\leq 1}$ is easy using the entries of T for B'_1, since $B'_1 = B'_{\leq 1}$ and so (taking the index shift of i into account) we have

$$\hat{T}[B'_{\leq 1}, k', (d_j^+, d_j^-)_{j=i}^{\lambda(W)}] = T[B'_1, k', (d_j^+, d_j^-)_{j=i}^{\lambda(W)}].$$

To compute entries of $\hat{T}$ for some $B'_{\leq \ell}$ where $\ell \geq 2$, we combine entries of table T for B'_ℓ with entries of table $\hat{T}$ for $B'_{\leq \ell - 1}$. However we will only combine entries with distance functions that imply compatible solutions. More concretely, we say that distance functions $(d_j^+, d_j^-)_{j=i}^{\lambda(W)}$ for $B'_{\leq \ell}$, $(\delta_j^+, \delta_j^-)_{j=i}^{\lambda(W)}$ for B'_ℓ, and $(\beta_j^+, \beta_j^-)_{j=i}^{\lambda(W)}$ for $B'_{\leq \ell-1}$ are *consistent* if for every level $j \geq i$ and $p \in I_B^j$ we have one of

1. $d_j^+(p) = \delta_j^+(p) = \beta_j^+(p)$ and $d_j^-(p) = \delta_j^-(p) = \beta_j^-(p) = \infty$, or
2. $d_j^-(p) = \delta_j^-(p) = \beta_j^+(p)$ and $d_j^+(p) = \delta_j^+(p) = \beta_j^-(p) = \infty$, or
3. $d_j^-(p) = \delta_j^+(p) = \beta_j^-(p)$ and $d_j^+(p) = \delta_j^-(p) = \beta_j^+(p) = \infty$.

The algorithm now considers all sets of consistent distance functions to compute an entry $\hat{T}[B'_{\leq \ell}, k', (d_j^+, d_j^-)_{j=i}^{\lambda(W)}]$ for $\ell \geq 2$ by setting it to

$$\begin{aligned} \min \big\{ T[B'_\ell, k'', (\delta_j^+, \delta_j^-)_{j=i}^{\lambda(W)}] + \hat{T}[B'_{\leq \ell-1}, k' - k'', (\beta_j^+, \beta_j^-)_{j=i}^{\lambda(W)}] \ \mid \\ k'' \in \{0, \ldots, k'\} \text{ and } (d_j^+, d_j^-)_{j=i}^{\lambda(W)}, (\delta_j^+, \delta_j^-)_{j=i}^{\lambda(W)}, (\beta_j^+, \beta_j^-)_{j=i}^{\lambda(W)} \text{ are consistent} \big\} \end{aligned} \quad (1)$$

We now prove the correctness using two lemmas. The following lemma implies that if we only consider consistent distance functions to compute entries recursively, then the entries will store values for compatible solutions.

Lemma 22. *Let $(d_j^+, d_j^-)_{j=i}^{\lambda(W)}$ for $B'_{\leq \ell}$, $(\delta_j^+, \delta_j^-)_{j=i}^{\lambda(W)}$ for B'_ℓ, and $(\beta_j^+, \beta_j^-)_{j=i}^{\lambda(W)}$ for $B'_{\leq \ell-1}$ be consistent distance functions, and let $S_1 = B'_\ell \cap F$ and $S_2 = B'_{\leq \ell-1} \cap F$ be facility sets. If S_1 is compatible with entry $T[B'_\ell, |S_1|, (\delta_j^+, \delta_j^-)_{j=i}^{\lambda(W)}]$ and S_2 is compatible with entry $\hat{T}[B'_{\leq \ell-1}, |S_2|, (\beta_j^+, \beta_j^-)_{j=i}^{\lambda(W)}]$, then the union $S = S_1 \cup S_2$ is compatible with entry $\hat{T}[B'_{\leq \ell}, |S|, (d_j^+, d_j^-)_{j=i}^{\lambda(W)}]$. Moreover, $\hat{C}_{\leq \ell}(S) = C_{B'_\ell}(S_1) + \hat{C}_{\leq \ell-1}(S_2)$.*

Proof. To prove compatibility of S with the entry $\hat{T}[B'_{\leq\ell}, |S|, (d_j^+, d_j^-)_{j=i}^{\lambda(W)}]$, it suffices to show that the distance functions are set correctly. Fix a level $j \geq i$ and an interface point $p \in I_B^j$. There are three cases to consider, according to the definition of consistency of the distance functions. In the first case, all three internal distance functions are set to ∞, and all external distance functions are set to the same value. In particular, since S_1 and S_2 are compatible with their respective entries, we have $d_j^+(p) = \delta_j^+(p) = \beta_j^+(p) \leq \min\{\langle \text{dist}(p, S_1)\rangle_j, \langle \text{dist}(p, S_2)\rangle_j\} = \langle \text{dist}(p, S)\rangle_j$, as $S = S_1 \cup S_2$. In the second case, $\beta_j^-(p) = \delta_j^+(p) = \infty$ and so $\beta_j^+(p) \leq \langle \text{dist}(p, S_2)\rangle_j$, since S_2 is compatible with its entry, and $\delta_j^-(p) = \langle \text{dist}(p, S_1)\rangle_j$, since S_1 is compatible with its entry. Since we also have $\beta_j^+(p) = \delta_j^-(p)$ we get $\langle \text{dist}(p, S_1)\rangle_j \leq \langle \text{dist}(p, S_2)\rangle_j$, and hence $\langle \text{dist}(p, S)\rangle_j = \langle \text{dist}(p, S_1)\rangle_j$. Consistency furthermore implies $d_j^-(p) = \delta_j^-(p) = \langle \text{dist}(p, S)\rangle_j$ and $d_j^+(p) = \infty$. The third case is analogous to the second, and therefore S is compatible with its entry.

For the second part, we consider the contributions of vertices to the terms $\hat{C}_{\leq\ell}(S)$, $C_{B'_\ell}(S_1)$, and $\hat{C}_{\leq\ell-1}(S_2)$, and show that they are the same for $\hat{C}_{\leq\ell}(S)$ and for $C_{B'_\ell}(S_1) + \hat{C}_{\leq\ell-1}(S_2)$. For this we first fix a vertex $v \in B'_{\leq\ell-1}$, and in the following distinguish the cases where its contribution to $\hat{C}_{\leq\ell-1}(S_2)$ and $\hat{C}_{\leq\ell}(S)$ is due to a facility or an interface point.

The first case is that $\text{dist}'(v, S_2) \leq \min_{j\geq i,\, p\in I_B^j}\{\text{dist}(v, p) + \beta_j^+(p)\}$, i.e., the contribution of v to $\hat{C}_{\ell-1}(S_2)$ is given by a facility of S_2. Note that the consistency of the distance functions always implies that $\beta_j^+(p) = d_j^+(p)$ or $d_j^+(p) = \infty$ for any level $j \geq i$ and interface point $p \in I^j$, and so $\min_{j\geq i,\, p\in I_B^j}\{\text{dist}(v, p) + \beta_j^+(p)\} \leq \min_{j\geq i,\, p\in I_B^j}\{\text{dist}(v, p) + d_j^+(p)\}$. At the same time $\text{dist}'(v, S) \leq \text{dist}'(v, S_2)$ as $S_2 \subseteq S$. We hence get that $\text{dist}'(v, S) \leq \min_{j\geq i,\, p\in I_B^j}\{\text{dist}(v, p) + d_j^+(p)\}$, i.e., the contribution of v to $\hat{C}_\ell(S)$ is also given by a facility of S in this case. Thus to show that the contribution of v to $\hat{C}_{\ell-1}(S_2)$ and $\hat{C}_\ell(S)$ is the same, we need to show that $\text{dist}'(v, S) = \text{dist}'(v, S_2)$. Note that this is implied if $\text{dist}'(v, S) \geq \min_{j\geq i,\, p\in I_B^j}\{\text{dist}(v, p) + \beta_j^+(p)\}$, since we have $\text{dist}'(v, S) \leq \text{dist}'(v, S_2) \leq \min_{j\geq i,\, p\in I_B^j}\{\text{dist}(v, p) + \beta_j^+(p)\}$. Thus the following proves the claim, using that the contribution of v to $\hat{C}_\ell(S)$ is given by a facility of S.

Claim 23. *For $v \in B'_{\leq\ell-1}$, if $\text{dist}'(v, S) \leq \min_{j\geq i,\, p\in I_B^j}\{\text{dist}(v, p) + d_j^+(p)\}$ then we have $\text{dist}'(v, S) = \text{dist}'(v, S_2)$ or $\text{dist}'(v, S) \geq \min_{j\geq i,\, p\in I_B^j}\{\text{dist}(v, p) + \beta_j^+(p)\}$.*

Proof. Given $\text{dist}'(v, S) \leq \min_{j\geq i,\, p\in I_B^j}\{\text{dist}(v, p) + d_j^+(p)\}$, assume to the contrary that we have $\text{dist}'(v, S) < \min_{j\geq i,\, p\in I_B^j}\{\text{dist}(v, p) + \beta_j^+(p)\}$ and $\text{dist}'(v, S) \neq \text{dist}'(v, S_2)$, which, as $S = S_1 \cup S_2$, means $\text{dist}'(v, S) < \text{dist}'(v, S_2)$. The latter inequality implies that the value of $\text{dist}'(v, S)$ is obtained for some facility $f \in S_1 \subseteq B'_\ell$. In particular, $v \in B'_{\leq\ell-1}$ and $f \in B'_\ell$ are cut at level $i - 1$, and so there is an interface point $p \in I_B^i$ such that $\text{dist}'(v, S) = \text{dist}(v, p) + \langle \text{dist}(p, f)\rangle_i$, and f is the closest facility to p in S, i.e, $\langle \text{dist}(p, S)\rangle_i = \langle \text{dist}(p, f)\rangle_i$. Using the former of the assumed inequalities we get $\text{dist}(v, p) + \langle \text{dist}(p, f)\rangle_i = \text{dist}'(v, S) < \text{dist}(v, p) + \beta_i^+(p)$, and so we can conclude that $\langle \text{dist}(p, f)\rangle_i < \beta_i^+(p)$.

Using the inequality of the premise of the claim, we also get $\text{dist}(v, p) + \langle \text{dist}(p, f)\rangle_i = \text{dist}'(v, S) \leq \text{dist}(v, p) + d_i^+(p)$, i.e. $\langle \text{dist}(p, f)\rangle_i \leq d_i^+(p)$. Since S is compatible with entry $\hat{T}[B'_{\leq\ell}, |S|, (d_j^+, d_j^-)_{j=i}^{\lambda(W)}]$, we have $d_i^+(p) = \infty$ or $d_i^+(p) \leq \langle \text{dist}(p, S)\rangle_i$. In the latter case we would have $d_i^+(p) \leq \langle \text{dist}(p, S)\rangle_i = \langle \text{dist}(p, f)\rangle_i < \beta_i^+(p)$, which however cannot happen if the distance functions are consistent. Thus compatibility of S implies $d_i^+(p) = \infty$ and $d_i^-(p) = \langle \text{dist}(p, f)\rangle_i$. In particular, we can conclude that $d_i^-(p)$ has a finite value (as f exists) and $\beta_i^+(p)$ differs from $d_i^-(p)$. This can only mean that the third of the consistency properties applies to p at level i, and so $\beta_i^-(p) = d_i^-(p) = \langle \text{dist}(p, f)\rangle_i$.

In particular, also $\beta_i^-(p)$ has a finite value, and using the compatibility of S_2 with entry $\hat{T}[B'_{\leq \ell-1}, |S_2|, (\beta_j^+, \beta_j^-)_{j=i}^{\lambda(W)}]$, we can conclude that there exists a facility $f' \in S_2 \subseteq B'_{\leq \ell-1}$ with $\langle \text{dist}(p, f') \rangle_i = \beta_i^-(p) = \langle \text{dist}(p, f) \rangle_i$. Now let $j \leq i$ be the level for which $v \in B'_{\leq \ell-1}$ and $f' \in B'_{\leq \ell-1}$ are cut at level $j-1$ by $\mathcal{D}$. Lemma 19 implies $\text{dist}'(v, f') \leq \text{dist}(v, p) + \langle \text{dist}(p, f') \rangle_i$, but then we have

$$\text{dist}'(v, S_2) \leq \text{dist}'(v, f') \leq \text{dist}(v, p) + \langle \text{dist}(p, f') \rangle_i = \text{dist}(v, p) + \langle \text{dist}(p, f) \rangle_i = \text{dist}'(v, S),$$

which is a contradiction to $\text{dist}'(v, S) < \text{dist}'(v, S_2)$. ⌟

The next case we consider is that $\min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + d_j^+(p)\} < \text{dist}'(v, S)$, i.e., the contribution of v to $\hat{C}_\ell(S)$ is given by an interface point. As observed before, we have $\min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + \beta_j^+(p)\} \leq \min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + d_j^+(p)\}$ and $\text{dist}'(v, S) \leq \text{dist}'(v, S_2)$, which implies $\min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + \beta_j^+(p)\} < \text{dist}'(v, S_2)$, i.e. in this case the contribution of v to $\hat{C}_{\ell-1}(S_2)$ is also given by an interface point. Note that it also implies $\text{dist}'(v, S) > \min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + \beta_j^+(p)\}$, and thus the following claim shows that the contribution of v to $\hat{C}_\ell(S)$ and $\hat{C}_{\ell-1}(S_2)$ is the same.

Claim 24. *For $v \in B'_{\leq \ell-1}$, if $\text{dist}'(v, S) > \min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + \beta_j^+(p)\}$ then we have $\min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + \beta_j^+(p)\} = \min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + d_j^+(p)\}$.*

Proof. Given $\text{dist}'(v, S) > \min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + \beta_j^+(p)\}$, assume to the contrary that $\min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + \beta_j^+(p)\} \neq \min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + d_j^+(p)\}$. As observed before, the consistency of the distance functions always implies $\beta_j^+(p) = d_j^+(p)$ or $d_j^+(p) = \infty$, and thus we must have $\min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + \beta_j^+(p)\} < \min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + d_j^+(p)\}$. Let $j \geq i$ and $p \in I_B^j$ be the level and interface point for which the minimum of the former term of this inequality is obtained. The inequality then implies $\beta_j^+(p) < d_j^+(p)$ for this particular point p and level j, which can only be the case if $\beta_j^+(p) < \infty$ and $d_j^+(p) = \infty$. The values of $\beta_j^+(p)$ and $d_j^+(p)$ can only differ if the second of the consistency properties applies to p at level j, and so $\beta_j^+(p) = d_j^-(p)$. Since $\beta_j^+(p) < \infty$, the compatibility of S with entry $\hat{T}[B'_{\leq \ell}, |S|, (d_j^+, d_j^-)_{j=i}^{\lambda(W)}]$, implies $\beta_j^+(p) = d_j^-(p) = \langle \text{dist}(p, S) \rangle_j$.

Now let $f \in S \subseteq B'_{\leq \ell}$ be the facility for which $\text{dist}'(v, S) = \text{dist}'(v, f)$ (which exists as $d_j^-(p) < \infty$). Let $j' \leq i$ be the level for which $v \in B'_{\leq \ell-1}$ and $f \in B'_{\leq \ell}$ are cut at level $j' - 1$ by $\mathcal{D}$. By Lemma 19 we have $\text{dist}'(v, f) \leq \text{dist}(v, p) + \langle \text{dist}(p, f) \rangle_j$, since $j' \leq j$ and the part $B \in \mathcal{B}_i$ containing v and f is itself contained in some part $\tilde{B} \in \mathcal{B}_j$ with $v, f \in \tilde{B}$ and $p \in I_{\tilde{B}}$. But then,

$$\text{dist}'(v, S) \leq \text{dist}(v, p) + \langle \text{dist}(p, f) \rangle_j = \text{dist}(v, p) + \langle \text{dist}(p, S) \rangle_j = \text{dist}(v, p) + \beta_j^+(p).$$

However the last term is equal to $\min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + \beta_j^+(p)\}$, which gives a contradiction to our premise $\text{dist}'(v, S) > \min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + \beta_j^+(p)\}$. ⌟

So far we considered the case when the contribution of v to $\hat{C}_{\ell-1}(S_2)$ is given by a facility, or when the contribution of v to $\hat{C}_\ell(S)$ is given by an interface point. Thus the last case we consider is when the contribution of v to $\hat{C}_{\ell-1}(S_2)$ is given by an interface point, and the contribution of v to $\hat{C}_\ell(S)$ is given by a facility, i.e., $\min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + \beta_j^+(p)\} < \text{dist}'(v, S_2)$ and $\text{dist}'(v, S) \leq \min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + d_j^+(p)\}$. We need to show that $\text{dist}'(v, S) = \min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + \beta_j^+(p)\}$. First assume $\text{dist}'(v, S) > \min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + \beta_j^+(p)\}$. Due to Claim 24 this would imply $\text{dist}'(v, S) > \min_{j \geq i,\, p \in I_B^j} \{\text{dist}(v, p) + d_j^+(p)\}$, which however contradicts our assumption to the contrary, i.e., that the contribution of v to $\hat{C}_\ell(S)$ is given by a facility. Hence we must instead have $\mathrm{dist}'(v,S) \le \min_{j\ge i,\, p\in I_B^j}\{\mathrm{dist}(v,p)+\beta_j^+(p)\}$.

According to Claim 23, our assumption that $\mathrm{dist}'(v,S) \le \min_{j\ge i,\, p\in I_B^j}\{\mathrm{dist}(v,p)+d_j^+(p)\}$ implies $\mathrm{dist}'(v,S) = \mathrm{dist}'(v,S_2)$ or $\mathrm{dist}'(v,S) \ge \min_{j\ge i,\, p\in I_B^j}\{\mathrm{dist}(v,p)+\beta_j^+(p)\}$. In the former case, together with our assumption that the contribution of v to $\hat{C}_{\ell-1}(S_2)$ is given by an interface point, we would get $\mathrm{dist}'(v,S) > \min_{j\ge i,\, p\in I_B^j}\{\mathrm{dist}(v,p)+\beta_j^+(p)\}$, for which we saw above that this leads to a contradiction via Claim 24. Hence we are left with the other implication of Claim 23, i.e., $\mathrm{dist}'(v,S) \ge \min_{j\ge i,\, p\in I_B^j}\{\mathrm{dist}(v,p)+\beta_j^+(p)\}$. This together with our conclusion from above, i.e., $\mathrm{dist}'(v,S) \le \min_{j\ge i,\, p\in I_B^j}\{\mathrm{dist}(v,p)+\beta_j^+(p)\}$, means that the contribution of v to $\hat{C}_\ell(S)$ and $\hat{C}_{\ell-1}(S_2)$ is the same.

By analogous arguments, the contribution of any $v \in B'_\ell$ to $C_{B'_\ell}(S_1)$ is the same as its contribution to $\hat{C}_{\le\ell}(S)$. Since B'_ℓ and $B'_{\le\ell-1}$ partition the set $B'_{\le\ell}$, this means that $\hat{C}_{\le\ell}(S) = C_{B'_\ell}(S_1) + \hat{C}_{\le\ell-1}(S_2)$, as required. □

The next lemma implies that the compatible facility set minimizing $\hat{C}_{\le\ell}(S)$ is considered as a solution when recursing over consistent distance functions.

Lemma 25. *Let $S = B'_{\le\ell}\cap F$ be a facility set of $B'_{\le\ell}$ that is compatible with entry $\hat{T}[B'_{\le\ell}, |S|, (d_j^+, d_j^-)_{j=i}^{\lambda(W)}]$, and let $S_1 = S\cap B'_\ell$ and $S_2 = S\cap B'_{\le\ell-1}$. Then there exist distance functions $(\delta_j^+,\delta_j^-)_{j=i}^{\lambda(W)}$ for B'_ℓ, and $(\beta_j^+,\beta_j^-)_{j=i}^{\lambda(W)}$ for $B'_{\le\ell-1}$ such that*

- *$(d_j^+, d_j^-)_{j=i}^{\lambda(W)}$, $(\delta_j^+,\delta_j^-)_{j=i}^{\lambda(W)}$, and $(\beta_j^+,\beta_j^-)_{j=i}^{\lambda(W)}$ are consistent, and*
- *the set S_1 is compatible with entry $T[B'_\ell, |S_1|, (\delta_j^+,\delta_j^-)_{j=i}^{\lambda(W)}]$ and S_2 is compatible with entry $\hat{T}[B'_{\le\ell-1}, |S_2|, (\beta_j^+,\beta_j^-)_{j=i}^{\lambda(W)}]$.*

Proof. Consider any interface point $p \in I_B^j$ on some level $j \ge i$. Since S is compatible with entry $\hat{T}[B'_{\le\ell}, |S|, (d_j^+, d_j^-)_{j=i}^{\lambda(W)}]$, we either have $d_j^+(p) = \infty$ and $d_j^-(p) = \langle\mathrm{dist}(p,S)\rangle_j$, or $d_j^+(p) \le \langle\mathrm{dist}(p,S)\rangle_j$ and $d_j^-(p) = \infty$. Note that $S = S_1\cup S_2$ means $\langle\mathrm{dist}(p,S)\rangle_j = \min\{\langle\mathrm{dist}(p,S_1)\rangle_j, \langle\mathrm{dist}(p,S_2)\rangle_j\}$. Hence if $d_j^+(p) \le \langle\mathrm{dist}(p,S)\rangle_j$ we may set $\beta_j^+(p) = \delta_j^+(p) = d_j^+(p)$ and $\beta_j^-(p) = \delta_j^-(p) = \infty$, and obtain the first case of the consistency properties. Observe that this also implies the compatibility property of S_1 and S_2 for p.

Now assume that $d_j^+(p) = \infty$ and $\langle\mathrm{dist}(p,S_1)\rangle_j \le \langle\mathrm{dist}(p,S_2)\rangle_j$. Then we may set $\delta_j^-(p) = \beta_j^+(p) = d_j^-(p)$ and $\delta_j^+(p) = \beta_j^-(p) = \infty$. Since $d_j^-(p) = \langle\mathrm{dist}(p,S)\rangle_j = \langle\mathrm{dist}(p,S_1)\rangle_j$, this gives the second case of the consistency properties. Again, this also implies the compatibility property of S_1 and S_2 for p.

The remaining case when $d_j^+(p) = \infty$ and $\langle\mathrm{dist}(p,S_1)\rangle_j > \langle\mathrm{dist}(p,S_2)\rangle_j$ is analogous. Here we may set $\delta_j^+(p) = \beta_j^-(p) = d_j^-(p)$ and $\delta_j^-(p) = \beta_j^+(p) = \infty$. Because $d_j^-(p) = \langle\mathrm{dist}(p,S)\rangle_j = \langle\mathrm{dist}(p,S_2)\rangle_j$, this gives the third case of the consistency properties, and also implies the compatibility property of S_1 and S_2 for p, which concludes the proof. □

To argue that the algorithm sets the value of $\hat{T}[B'_{\le\ell}, k', (d_j^+, d_j^-)_{j=i}^{\lambda(W)}]$ correctly via (1), consider a set $S \subseteq B'_{\le\ell}$ that is compatible with this entry and minimizes $\hat{C}_{\le\ell}(S)$. By induction, Lemma 25 implies $T[B'_\ell, |S_1|, (\delta_j^+,\delta_j^-)_{j=i}^{\lambda(W)}] \le C_{B'_\ell}(S_1)$ and $\hat{T}[B'_{\le\ell-1}, |S_2|, (\beta_j^+,\beta_j^-)_{j=i}^{\lambda(W)}] \le \hat{C}_{\le\ell-1}(S_2)$, where $S_1 = S\cap B'_\ell$ and $S_2 = S\cap B'_{\le\ell-1}$. From (1) we therefore obtain $\hat{T}[B'_{\le\ell}, |S|, (d_j^+, d_j^-)_{j=i}^{\lambda(W)}] \le$

$C_{B'_\ell}(S_1) + \hat{C}_{\leq \ell-1}(S_2)$. By Lemma 22 only compatible sets are stored in an entry by induction, and so the definition of S implies $\hat{T}[B'_{\leq\ell}, |S|, (d_j^+, d_j^-)_{j=i}^{\lambda(W)}] = \hat{C}_{\leq\ell}(S)$, as required.

Bounding the runtime. To bound the size of the tables T and $\hat{T}$, note that since there are $\lambda(W) - \xi(W) + 1 \leq 2\log_2(nX/\varepsilon) + 2$ considered levels i, and each level $\mathcal{B}_i$ of $\mathcal{D}$ is a partition of W where $|W| \leq n$, there are at most $O(n \log(nX/\varepsilon))$ parts B considered by T in total. The other table $\hat{T}$ considers the same number of parts, since a set $B'_{\leq\ell}$ can be uniquely mapped to the part B'_ℓ. The number of possible values for k' is $k+1 = O(n)$. The domain $\{\langle x\rangle_j \mid 0 < x \leq 2^{j+6}\} \cup \{\infty\}$ of a distance function for level j has at most $\lceil 2^{j+6}/(\rho 2^j)\rceil + 1 = O(1/\rho)$ values, since $\langle x\rangle_j$ rounds a value to a multiple of $\rho 2^j$. The conciseness of the interface sets means that $|I_B^j| \leq (h/\rho)^{O(1)}$ according to Lemma 5. Hence there are at most $O(1/\rho)^{(h/\rho)^{O(1)}} = 2^{(h/\rho)^{O(1)}}$ possible distance functions. Since each entry of the table stores two distance functions for each of at most $2\log_2(nX/\varepsilon) + 2$ levels, the total number of entries of T and $\hat{T}$ is at most

$$O(n \log(nX/\varepsilon)) \cdot n \cdot (2^{(h/\rho)^{O(1)}})^{O(\log(nX/\varepsilon))} = (nX/\varepsilon)^{(h/\rho)^{O(1)}}.$$

Computing an entry of a table is dominated by (1). Going through all values $k' \leq n$ and all possible consistent distance functions to compute (1), takes $n \cdot 2^{(h/\rho)^{O(1)}}$ time, as there are $2^{(h/\rho)^{O(1)}}$ possible distance functions. Hence the total runtime is $(nX/\varepsilon)^{(h/\rho)^{O(1)}}$, proving Lemma 21.

The Facility Location^q problem. To compute an optimum rounded interface-respecting solution to Facility Location^q, the tables T and $\hat{T}$ can ignore the number of open facilities k', i.e., they have respective entries $T[B, (d_j^+, d_j^-)_{j=i+1}^{\lambda(W)}]$ and $\hat{T}[B'_{\leq\ell}, (d_j^+, d_j^-)_{j=i}^{\lambda(W)}]$. Accordingly, compatibility of facility sets with entries is defined as before, but ignoring the sizes of the sets. The value stored in each entry now also takes the opening costs of facilities into account. That is, for any set of facilities $S \subseteq F \cap B$ in a part B we define

$$C_B(S) = \sum_{v\in B} \chi_{\mathcal{I}_0}(v) \cdot \min\Big\{ \operatorname{dist}'(v,S), \min_{\substack{j\geq i+1\\ p\in I_B^j}} \{ \operatorname{dist}(v,p) + d_j^+(p)\}\Big\} + \sum_{f\in S} w_f,$$

and an entry $T[B, (d_j^+, d_j^-)_{j=i+1}^{\lambda(W)}]$ stores the minimum value of $C_B(S)$ over all sets S compatible with the entry, or ∞ if no such set exists. For $S \subseteq F \cap B'_{\leq\ell}$ in a union of subparts $B'_{\leq\ell}$ we define

$$\hat{C}_{\leq\ell}(S) = \sum_{v\in B'_{\leq\ell}} \chi_{\mathcal{I}_0}(v) \cdot \min\Big\{ \operatorname{dist}'(v,S), \min_{\substack{j\geq i\\ p\in I_B^j}} \{ \operatorname{dist}(v,p) + d_j^+(p)\}\Big\} + \sum_{f\in S} w_f,$$

and an entry $\hat{T}[B'_{\leq\ell}, (d_j^+, d_j^-)_{j=i}^{\lambda(W)}]$ stores the minimum value of $\hat{C}_{\leq\ell}(S)$ over all sets S compatible with the entry, or ∞ if no such set exists.

The entries of the tables can be computed in the same manner as before, but ignoring the set sizes. In particular, the most involved recursion becomes

$$\hat{T}[B'_{\leq\ell}, (d_j^+, d_j^-)_{j=i}^{\lambda(W)}] = \min\big\{ T[B'_\ell, (\delta_j^+, \delta_j^-)_{j=i}^{\lambda(W)}] + \hat{T}[B'_{\leq\ell-1}, (\beta_j^+, \beta_j^-)_{j=i}^{\lambda(W)}] \;\mid\; (d_j^+, d_j^-)_{j=i}^{\lambda(W)}, (\delta_j^+, \delta_j^-)_{j=i}^{\lambda(W)}, (\beta_j^+, \beta_j^-)_{j=i}^{\lambda(W)} \text{ are consistent}\big\}.$$

Note that if $S_1 = B'_\ell \cap F$ and $S_2 = B'_{\leq\ell-1} \cap F$ then these two sets are disjoint, and so $\sum_{f\in S} w_f = \sum_{f\in S_1} w_f + \sum_{f\in S_2} w_f$ for the union $S = S_1 \cup S_2$. Hence when proving $\hat{C}_{\leq\ell}(S) = C_{B'_\ell}(S_1) + \hat{C}_{\leq\ell-1}(S_2)$ for Lemma 22, we can ignore the facility opening costs, and the proof remains the same as before. All other arguments carry over, and thus an optimum rounded interface-respecting solution for an instance of Facility Location^q can also be computed in $(nX/\varepsilon)^{(h/\rho)^{O(1)}}$ time.

5 Hardness for graphs of highway dimension 1

For both k-Clustering^q and Facility Location^q we present the same reduction from the NP-hard satisfiability problem (SAT), in which a boolean formula φ in conjunctive normal form is given, and a satisfying assignment of its variables needs to be found.

For a given SAT formula φ with k variables and ℓ clauses we construct a graph G_φ as follows. For each variable x we introduce a path $P_x = (t_x, u_x, f_x)$ with two edges of length 1 each. The two endpoints t_x and f_x are facilities of F and the additional vertex u_x is a client, i.e., $\chi(u_x) = 1$. For each clause C_i, where $i \in [\ell]$, we introduce a vertex v_i and add the edge $v_i t_x$ for each variable x such that C_i contains x as a positive literal, and we add the edge $v_i f_x$ for each x for which C_i contains x as a negative literal. Every edge incident to v_i has length $(11c)^i$ for the constant $c > 4$ due to Definition 1, and v_i is also a client, i.e., $\chi(v_i) = 1$. In case of Facility Location^q, every facility $f \in F$ has cost $w_f = 1$, i.e., we construct an instance of the uniform version of the problem.

Lemma 26. *The constructed graph G_φ has highway dimension* 1.

Proof. Fix a scale $r > 0$ and let $i = \lfloor \log_{11c}(r/5) + 1 \rfloor$. Note that $\beta_w(cr)$ cannot contain any edge incident to a vertex v_j for $j \geq i+1$, since the length of every such edge is $(11c)^j \geq 11cr/5 > 2cr$ and the diameter of $\beta_w(cr)$ is at most $2cr$. Thus if $\beta_w(cr)$ contains a vertex v_j for $j \geq i+1$, then $\beta_w(cr)$ contains only v_j, and there is nothing to prove. Note also that any path in $\beta_w(cr)$ that does not use v_i has length at most $2 + \sum_{j=1}^{i-1}(2(11c)^j + 2)$, since any such path can contain at most two edges incident to a vertex v_j and the paths P_x of length 2 are connected only through edges incident to vertices v_j. The length of such a path is thus strictly shorter than

$$2 + 2\left(\frac{(11c)^i}{11c-1} - 1\right) + 2i \leq 5(11c)^{i-1} \leq r,$$

where the first inequality holds since $i \geq 1$ and $c > 4$. Hence the only paths that need to be hit by hubs on scale r are those passing through v_i, which can clearly be done using only one hub, namely v_i. □

To finish the reduction for k-Clustering^q, we claim that there is a satisfying assignment for φ if and only if there is a solution for G_φ with cost at most $k + \sum_{i=1}^{\ell}(11c)^{iq}$. If there is a satisfying assignment for φ we open each facility t_x for variables x that are set to true, and we open each facility f_x for variables x that are set to false. This opens exactly k facilities and the cost of the solution is $k + \sum_{i=1}^{\ell}(11c)^{iq}$, since each of the k vertices u_x is assigned to either t_x or f_x at distance 1, and vertex v_i is assigned to a vertex t_x or f_x at distance $(11c)^i$ that corresponds to a literal of C_i that is true.

Conversely, assume there is a solution to k-Clustering^q of cost at most $k + \sum_{i=1}^{\ell}(11c)^{iq}$ in G_φ. Note that the minimum distance from any u_x to a facility is 1, while the minimum distance from any v_i to a facility is $(11c)^i$. Thus any solution must have cost at least $k + \sum_{i=1}^{\ell}(11c)^{iq}$, so that the assumed solution must open a facility at minimum distance for each client of G_φ. In particular, for each variable x, at least one of the facilities t_x and f_x is opened by the solution. Moreover, as only k facilities can be opened and there are k variables, exactly one of t_x and f_x is opened for each x. Thus the k-Clustering^q solution in G_φ can be interpreted as an assignment for φ, where we set a variable x to true if t_x is opened, and we set it to false if f_x is opened. Since also for each v_i the solution opens a facility at minimum distance, there must be a variable in C_i that is set so that its literal in C_i is true, i.e., the assignment satisfies φ. Thus due to the above lemma bounding the highway dimension of G_φ, we obtain the Theorem 3 for k-Clustering^q.

For Facility Location^q we claim that there is a satisfying assignment for φ if and only if there is a solution for G_φ of cost at most $2k + \sum_{i=1}^{\ell}(11c)^{iq}$. In fact the arguments are exactly the same as for k-CLUSTERING^q above: if there is a satisfying assignment then a solution for FACILITY LOCATION^q of cost $2k + \sum_{i=1}^{\ell}(11c)^{iq}$ exists, by opening the k facilities corresponding to the assignment of cost 1 each. Conversely, any solution has cost at least $k + \sum_{i=1}^{\ell}(11c)^{iq}$ due to the edge lengths, and at least k facilities need to be opened, one for each variable gadget. This gives a minimum cost of $2k + \sum_{i=1}^{\ell}(11c)^{iq}$, and any such solution corresponds to a satisfying assignment of φ. This proves Theorem 3 for uniform FACILITY LOCATION^q.

References

[1] I. Abraham, D. Delling, A. Fiat, A. V. Goldberg and R. F. Werneck. 'Highway dimension and provably efficient shortest path algorithms'. *Journal of the ACM* 63.5 (2016), p. 41.

[2] I. Abraham, D. Delling, A. Fiat, A. Goldberg and R. Werneck. 'VC-dimension and shortest path algorithms'. In: *ICALP*. 2011, pp. 690–699.

[3] I. Abraham, A. Fiat, A. V. Goldberg and R. F. Werneck. 'Highway dimension, shortest paths, and provably efficient algorithms'. In: *SODA*. 2010, pp. 782–793.

[4] S. Ahmadian, A. Norouzi-Fard, O. Svensson and J. Ward. 'Better guarantees for k-means and euclidean k-median by primal-dual algorithms'. *SIAM Journal on Computing* 0 (2019), FOCS17–97.

[5] S. Arora, P. Raghavan and S. Rao. 'Approximation Schemes for Euclidean K-medians and Related Problems'. In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. STOC '98. Dallas, Texas, USA: ACM, 1998, pp. 106–113.

[6] P. Awasthi, M. Charikar, R. Krishnaswamy and A. K. Sinop. 'The Hardness of Approximation of Euclidean k-Means'. In: *31st International Symposium on Computational Geometry, SoCG 2015, June 22-25, 2015, Eindhoven, The Netherlands*. 2015, pp. 754–767.

[7] Y. Bartal and L.-A. Gottlieb. 'A Linear Time Approximation Scheme for Euclidean TSP'. In: *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS*. 2013.

[8] A. Becker, P. N. Klein and D. Saulpic. 'Polynomial-Time Approximation Schemes for k-Center and Bounded-Capacity Vehicle Routing in Metrics with Bounded Highway Dimension'. In: *ESA*. 2018, 8:1–8:15.

[9] J. Blum. 'Hierarchy of transportation network parameters and hardness results'. In: *IPEC*. 2019, 4:1–4:15.

[10] V. Braverman, S. H.-C. Jiang, R. Krauthgamer and X. Wu. 'Coresets for Clustering in Excluded-minor Graphs and Beyond'. *arXiv preprint arXiv:2004.07718* (2020).

[11] J. Byrka, T. Pensyl, B. Rybicki, S. Aravind and K. Trinh. 'An Improved Approximation for k-median, and Positive Correlation in Budgeted Optimization'. In: *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*. 2015, pp. 737–756.

[12] V. Cohen-Addad, A. E. Feldmann and D. Saulpic. 'Near-Linear Time Approximations Schemes for Clustering in Doubling Metrics'. In: *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS*. 2019, pp. 540–559.

[13] V. Cohen-Addad, P. N. Klein and C. Mathieu. 'Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics'. *SIAM Journal on Computing* 48.2 (2019), pp. 644–667.

[14] V. Cohen-Addad and C. Schwiegelshohn. ‘On the Local Structure of Stable Clustering Instances’. In: *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*. 2017, pp. 49–60.

[15] Y. Disser, A. E. Feldmann, M. Klimm and J. Könemann. ‘Travelling on Graphs with Small Highway Dimension’. In: *Graph-Theoretic Concepts in Computer Science - 45th International Workshop, WG*. Vol. 11789. Springer, 2019, pp. 175–189.

[16] A. E. Feldmann. ‘Fixed-Parameter Approximations for k-Center Problems in Low Highway Dimension Graphs’. *Algorithmica* 81.3 (2019), pp. 1031–1052.

[17] A. E. Feldmann, W. S. Fung, J. Könemann and I. Post. ‘A $(1+\varepsilon)$-Embedding of Low Highway Dimension Graphs into Bounded Treewidth Graphs’. *SIAM Journal on Computing* 47.4 (2018), pp. 1275–1734.

[18] A. E. Feldmann and D. Saulpic. ‘Polynomial Time Approximation Schemes for Clustering in Low Highway Dimension Graphs’. In: *28th Annual European Symposium on Algorithms ESA*. 2020, 46:1–46:22.

[19] Z. Friggstad, M. Rezapour and M. R. Salavatipour. ‘Local Search Yields a PTAS for k-Means in Doubling Metrics’. *SIAM J. Comput.* 48.2 (2019), pp. 452–480.

[20] S. Guha and S. Khuller. ‘Greedy strikes back: Improved facility location algorithms’. *Journal of algorithms* 31.1 (1999), pp. 228–248.

[21] A. Gupta, R. Krauthgamer and J. R. Lee. ‘Bounded Geometries, Fractals, and Low-Distortion Embeddings’. In: *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*. FOCS ’03. 2003.

[22] A. Gupta and K. Tangwongsan. ‘Simpler Analyses of Local Search Algorithms for Facility Location’. *CoRR* abs/0809.2554 (2008). arXiv: 0809.2554.

[23] K. Jain, M. Mahdian and A. Saberi. ‘A new greedy approach for facility location problems’. In: *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. 2002, pp. 731–740.

[24] S. G. Kolliopoulos and S. Rao. ‘A nearly linear-time approximation scheme for the Euclidean k-median problem’. *SIAM Journal on Computing* 37.3 (2007), pp. 757–782.

[25] S. Li. ‘A 1.488 approximation algorithm for the uncapacitated facility location problem’. *Inf. Comput.* 222 (2013), pp. 45–58.

[26] K. Talwar. ‘Bypassing the embedding: algorithms for low dimensional metrics’. In: *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*. ACM. 2004, pp. 281–290.

A Highway dimension vs doubling dimension

We discuss here the relationship between low doubling and low highway dimension metrics. First off, in a follow-up paper to [3], Abraham et al. [1] define a version of the highway dimension, which implies that the graphs also have bounded doubling dimension. Hence for this definition, the algorithm of Cohen-Addad et al. [12] is already a very efficient approximation scheme in metrics of low highway dimension. Definition 1 on the other hand implies metrics of large doubling dimension as noted by Abraham et al. [3]: a star has highway dimension 1 (by using the center vertex to hit all paths), but its doubling dimension is unbounded. While it may be reasonable to assume that road networks have low doubling dimension (which are the main concern in the works of Abraham et al. [1, 2, 3]), there are metrics modeling transportation networks, for which it can be argued that the doubling dimension is large, while the highway dimension should be small, and thus rather adhere to Definition 1: in networks arising from public transportation, longer connections are serviced by larger and and sparser stations (such as train stations and airports). More concretely, the so-called hub-and-spoke networks that can typically be seen in air traffic networks is much closer to a star-like network and is unlikely to have small doubling dimension, while still having small highway dimension. Thus in these examples it is reasonable to assume that the doubling dimension is a lot larger than the highway dimension.

For further discussions on different definitions of the highway dimension we refer to the work of Blum [9] and Section 9 of Feldmann et al. [17].

Low doubling and low highway dimension metrics still have some similarities, which are exploited for the results of this and other papers. However, as pointed out in the introduction, a major difference is the non-existence of an analogue to portal-respecting paths in low highway dimension metrics. We illustrate this by the example given in Fig. 2: the given graph contains a rooted tree (thick black edges) of height ℓ (in the picture, $\ell = 3$) for which each internal vertex at distance i from the root (topmost vertex) is connected to Δ children via edges of length $(11c)^{\ell-i}$, where $c > 4$ is the constant from Definition 1. Additionally, each leaf of the tree is adjacent to the root via a (thin black) edge of length $(11c)^{\ell}$. The resulting graph has doubling dimension $\log_2(\Delta)$, while the highway dimension is 1 (using a similar proof as for Lemma 26). The circles represent towns of the

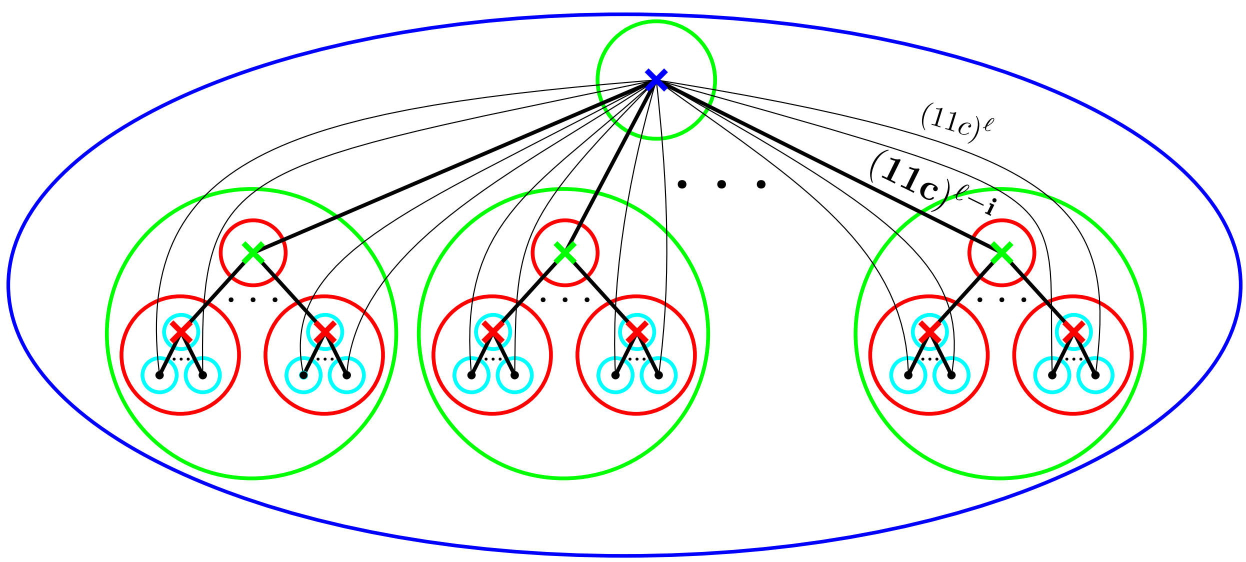

Figure 2: An example of a low highway dimension graph without an analogue to portal-respecting paths.

town decomposition, each colour being a different level of the decomposition. For each town T its set X_T contains one hub (crosses) with the same colour (e.g., each green town contains a green hub connecting its red child towns), except for the towns on the lowest level (light blue) where $X_T = \emptyset$.

In this example any portal set according to Lemma 4 will not be of constant size, since the doubling dimension is unbounded. Thus we want to exploit the towns and their hub sets instead. But then no analogue to near-optimal portal-respecting paths exists: connecting a leaf to the root through hubs of increasing levels results in the path through the tree (thick edges). This path has length $\sum_{i=0}^{\ell-1}(11c)^{\ell-i} > (11c)^{\ell} + (11c)^{\ell-1}$, while using the direct (thin) edge is shorter by a factor of more than $1 + \frac{1}{11c}$. Thus for sufficiently small ε, to obtain a near-optimal connection we must use the direct edge, i.e., the leaf and root need to be connected using the interface point (the dark blue hub) of the town containing them.

B Proof of Lemma 9

The first inequality follows immediately from [17, Lemma 3.2]. For the second inequality, we note that in [17] the towns of the town decomposition $\mathcal{T}$ are defined with respect to exponentially growing values $r_i = (c/4)^i$ where $c > 4$ is the constant of Definition 1. Here the index $i \in \mathbb{N}_0$ is called a *level*, but these levels behave quite differently from the levels of a hierarchical decomposition, which is why we refrained from introducing levels of town decompositions in this paper. In particular, the child towns of a town of level i might not be from level $i-1$, but can be from other levels as well. By [17, Lemma 3.3] however, the level of a child town of any town of level i is at most $i-1$. If the level of a town T is i, then by [17, Lemma 3.2] we have $\mathrm{diam}(T) \leq r_i$. Now let i be the level for which $\mathrm{diam}(T_0) \in (r_{i-1}, r_i]$ for a given town T_0. By the above properties, the level of a g^{th}-generation descendant T_g of T_0 is at most $i-g$, and so $\mathrm{diam}(T_g) \leq r_{i-g}$. Since $\mathrm{diam}(T_0) \geq r_{i-1}$ this implies the required bound $\mathrm{diam}(T_g) \leq \mathrm{diam}(T_0)/2^{g-1}$, if we set $c = 8$. □