eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Fixed-Parameter Approximations for k-Center Problems in Low Highway Dimension Graphs[*]

Andreas Emil Feldmann

KAM, Charles University, Czechia
`feldmann.a.e@gmail.com`

Abstract

We consider the k-CENTER problem and some generalizations. For k-CENTER a set of k *center vertices* needs to be found in a graph G with edge lengths, such that the distance from any vertex of G to its nearest center is minimized. This problem naturally occurs in transportation networks, and therefore we model the inputs as graphs with bounded *highway dimension*, as proposed by Abraham et al. [SODA 2010].

We show both approximation and fixed-parameter hardness results, and how to overcome them using *fixed-parameter approximations*, where the two paradigms are combined. In particular, we prove that for any $\varepsilon > 0$ computing a $(2 - \varepsilon)$-approximation is W[2]-hard for parameter k, and NP-hard for graphs with highway dimension $O(\log^2 n)$. The latter does not rule out fixed-parameter $(2 - \varepsilon)$-approximations for the highway dimension parameter h, but implies that such an algorithm must have at least doubly exponential running time in h if it exists, unless ETH fails. On the positive side, we show how to get below the approximation factor of 2 by combining the parameters k and h: we develop a fixed-parameter $3/2$-approximation with running time $2^{O(kh \log h)} \cdot n^{O(1)}$. Additionally we prove that, unless P=NP, our techniques cannot be used to compute fixed-parameter $(2 - \varepsilon)$-approximations for only the parameter h.

We also provide similar fixed-parameter approximations for the WEIGHTED k-CENTER and $(k, \mathcal{F})$-PARTITION problems, which generalize k-CENTER.

1 Introduction

In this paper we consider the k-CENTER problem and some of its generalizations. For the problem, k locations need to be found in a network, so that every node in the network is close to a location. More formally, the input is specified by an integer $k \in \mathbb{N}$ and a graph $G = (V, E)$ with positive edge lengths. A *feasible solution* to the problem is a set $C \subseteq V$ of *centers* such that $|C| \leq k$. The aim is to minimize the maximum distance between any vertex and its closest center. That is, let $\mathrm{dist}_G(u, v)$ denote the shortest-path distance between two vertices $u, v \in V$ of G according to the edge lengths, and $B_v(r) = \{u \in V \mid \mathrm{dist}_G(u, v) \leq r\}$ be the *ball of radius r around v*. We need to minimize the *cost* of the solution C, which is the smallest value ρ for which $\bigcup_{v \in C} B_v(\rho) = V$.

We say that a center $v \in C$ *covers* a vertex $u \in V$ if $u \in B_v(\rho)$. Hence we can see the problem as finding k centers covering all vertices of G with balls of minimum radius.

The k-CENTER problem naturally arises in transportation networks, where, for instance, it models the need to find locations for manufacturing plants, hospitals, police stations, or warehouses under a budget constraint. Unfortunately it is NP-hard to solve the problem in general [29], and the same holds true in various models for transportation networks, such as planar graphs [28] and metrics using Euclidean (L_2), Manhattan (L_1), or Chebyshev (L_∞) distance measures [14]. A more recent model for transportation networks uses the *highway dimension*, which was introduced as a graph parameter by Abraham et al. [1]. The intuition behind its definition comes from the empirical observation [7, 8] that in a road network, starting from any point A and travelling to a sufficiently far point B along the quickest route, one is bound to pass through some member of a sparse set of "access points". There are several formal definitions for the highway dimension that differ slightly [1, 2, 3, 16]. All of them, however, imply the existence of *locally sparse shortest path covers*. Therefore, in this paper we consider this as a generalization of the original highway dimension definitions.

Definition 1. Given a graph $G = (V, E)$ with edge lengths and a *scale* $r \in \mathbb{R}^+$, let $\mathcal{P}_{(r,2r]} \subseteq 2^V$ contain all vertex sets given by shortest paths in G of length more than r and at most $2r$. A *shortest path cover* $\text{SPC}(r) \subseteq V$ is a hitting set for the set system $\mathcal{P}_{(r,2r]}$, i.e., $P \cap \text{SPC}(r) \neq \emptyset$ for each $P \in \mathcal{P}_{(r,2r]}$. We call the vertices in $\text{SPC}(r)$ *hubs*. A hub set $\text{SPC}(r)$ is called *locally h-sparse* if for every vertex $v \in V$ the ball $B_v(2r)$ of radius $2r$ around v contains at most h vertices from $\text{SPC}(r)$. The *highway dimension* of G is the smallest integer h such that there is a locally h-sparse shortest path cover $\text{SPC}(r)$ for every scale $r \in \mathbb{R}^+$ in G.

Abraham et al. [1] introduced the highway dimension in order to explain the fast running times of various shortest-path heuristics. However, they also note that "conceivably, better algorithms for other [optimization] problems can be developed and analysed under the small highway dimension assumption". In this paper we investigate the k-CENTER problem and focus on graphs with low highway dimension as a model for transportation networks. One advantage of using such graphs is that they do not only capture road networks but also networks with transportation links given by air-traffic or railroads. For instance, introducing connections due to airplane traffic will render a network non-planar, while it can still be argued to have low highway dimension: longer flight connections tend to be served by bigger but sparser airports, which act as hubs. This can, for instance, be of interest in applications where warehouses need to be placed to store and redistribute goods of globally operating enterprises. Unfortunately however, in this paper we show that the k-CENTER problem also remains NP-hard on graphs with low highway dimension.

Two popular and well-studied ways of coping with NP-hard problems is to devise *approximation* [29, 30] and *parameterized* [11, 13] algorithms. For the former we demand polynomial running times but allow the computed solution to deviate from the optimum cost. That is, we compute a *c-approximation*, which is a feasible solution with a cost that is at most c times worse than the best possible for the given instance. A problem that allows a polynomial-time c-approximation for any input is *c-approximable*, and c is called the *approximation factor* of the corresponding algorithm. The rationale behind parameterized algorithms is that some *parameter p* of the input is small and we can therefore afford running times that are super-polynomial in p, while, however, we demand optimum solutions. That is, we compute a solution with optimum cost in time $f(p) \cdot n^{O(1)}$ for some computable function $f(\cdot)$ that is independent of the input size n. A problem that has a fixed-parameter algorithm for a parameter p is called *fixed-parameter tractable (FPT)* for p. What however, if a problem is neither approximable nor FPT? In this case it may be possible to overcome the complexity by combining these two paradigms. In particular, the objective becomes

to develop *fixed-parameter c-approximation* (*c*-FPA) algorithms that compute a *c*-approximation in time $f(p) \cdot n^{O(1)}$ for a parameter p.

The idea of combining the paradigms of approximation and fixed-parameter tractability has been suggested before. However, only few results are known for this setting (cf. [26]). In this paper we show that for the *k*-CENTER problem it is possible to overcome lower bounds for its approximability and its fixed-parameter tractability using parameterized approximations. For many different input classes, such as planar graphs [28], and L_1- and L_∞-metrics [14], the *k*-CENTER problem is 2-approximable via the algorithm for general metrics of Hochbaum and Shmoys [19], but not $(2 - \varepsilon)$-approximable for any $\varepsilon > 0$, unless P=NP. We show that, unless FPT=W[2], for general graphs there is no $(2 - \varepsilon)$-FPA algorithm for the parameter k. Additionally, we prove that, unless P=NP, *k*-CENTER is not $(2 - \varepsilon)$-approximable on graphs with highway dimension $O(\log^2 n)$. This does not rule out $(2 - \varepsilon)$-FPA algorithms for the highway dimension parameter, and we leave this as an open problem. However, the result implies that if such an algorithm exists, then its running time must be enormous. In particular, unless the exponential time hypothesis (ETH) [21, 22] fails, there can be no $(2 - \varepsilon)$-FPA algorithm with doubly exponential $2^{2^{o(\sqrt{h})}} \cdot n^{O(1)}$ running time in the highway dimension h.

In face of these hardness results, it seems tough to beat the approximation factor of 2 for *k*-CENTER, even when considering fixed-parameter approximations for either the parameter k or the highway dimension. Our main result, however, is that we can obtain a significantly better approximation factor for *k*-CENTER when combining these two parameters. Such an algorithm is useful when aiming for high quality solutions, for instance, in a setting where only few warehouses should be built in a transportation network, since warehouses are expensive or stored goods should not be too dispersed for logistical reasons.

It is known [2] that locally $O(h \log h)$-sparse shortest path covers can be computed for graphs of highway dimension h in polynomial time, if each shortest path is unique. We will assume that the latter is always the case, since we can slightly perturb the edge lengths. In particular, using a folklore method we may distort distances such that any 3/2-approximation in the perturbed instance also is a 3/2-approximation in the original instance. In the following theorem summarizing our main result, the first given running time assumes approximate shortest path covers. In general it is NP-hard to compute the highway dimension [16], but it is unknown whether this problem is FPT. If this is the case and the running time is sufficiently small, this can be used as an oracle in our algorithm.

Theorem 2. *For any graph G with n vertices and highway dimension h, there is an algorithm that computes a 3/2-approximation to the k-CENTER problem in time $2^{O(kh \log h)} \cdot n^{O(1)}$. If locally h-sparse shortest path covers are given by an oracle, the running time is $3^{kh} \cdot n^{O(1)}$.*

We leave open whether approximation factors better than 3/2 can be obtained for the combined parameter (k, h). It was recently proved [15] that *k*-CENTER is W[1]-hard for this parameter (k, h), but no inapproximability is implied by this result. We note that a recent result by Becker et al. [10] obtains a *fixed-parameter approximation scheme* for *k*-CENTER in low highway dimension graphs, i.e., an algorithm computing a $(1+\varepsilon)$-approximation in time $f(k, h, \varepsilon) \cdot n^{O(1)}$ for any $\varepsilon > 0$. However, this result needs a more restrictive definition of the highway dimension than used in this paper. In particular, there are graphs that have bounded highway dimension due to Definition 1, but for which the algorithm by Becker et al. [10] is not applicable (for a more detailed discussion on the relation between different definitions of highway dimension we refer to [16, Section 9]). Although we also leave open whether $(2 - \varepsilon)$-FPA algorithms exist for the parameter h alone, we are able to prove that the techniques we use for Theorem 2 cannot omit using both k and h as parameters. To obtain

a $(2 − ε)$-FPA algorithm with running time $f(h) · n^{O(1)}$ for any function $f(·)$ independent of n, a lot more information of the input would need to be exploited than the algorithm of Theorem 2 does. To explain this, we now turn to the used techniques.

1.1   Used techniques

A crucial observation for our algorithm is that at any scale r, a graph of low highway dimension is structured in the following way (see Figure 1). We will prove that the vertices are either at distance at most r from some hub, or they lie in clusters of diameter at most r that are at distance more than $2r$ from each other. Hence, for the cost $ρ$ of the optimum k-CENTER solution, at scale $r = ρ/2$ a center that resides in a cluster cannot cover any vertices of some other cluster. In this sense the clusters are "independent" of each other. At the same time we are able to bound the number of hubs of scale $ρ/2$ in terms of k and the highway dimension. Roughly, this is comparable to graphs with small vertex cover, since the vertices that are not part of a vertex cover form an independent set. In this
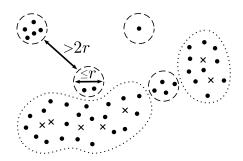

Figure 1:   Clusters (dashed circles) are far from hubs (crosses). They have small diameter and are far from each other.

sense the highway dimension is a generalization of the vertex cover number (this is in fact the reason why computing the highway dimension is NP-hard [16]).

At the same time the k-CENTER problem is a generalization of the DOMINATING SET problem. This problem is W[2]-hard [13], which, as we will show, is also why k-CENTER is W[2]-hard to approximate for parameter k. However, DOMINATING SET is FPT using the vertex cover number as the parameter [5]. This is one of the reasons why combining the two parameters k and h yields a $3/2$-FPA algorithm for k-CENTER. In fact the similarity seems so striking at first that one is tempted to reduce the problem of finding a $3/2$-approximation for k-CENTER on low highway dimension graphs to solving DOMINATING SET on a graph of low vertex cover number. However, it is unclear how this can be made to work. Instead we devise an involved algorithm that is driven by the intuition that the two problems are similar.

The algorithm will guess the cost $ρ$ of the optimum solution in order to exploit the structure of the graph given by the locally h-sparse shortest path cover for scale $r = ρ/2$. In particular, the shortest path covers of other scales do not need to be locally sparse in order for the algorithm to succeed. We will show that there are graphs for which k-CENTER is not $(2 − ε)$-approximable, unless P=NP, and for which the shortest path cover for scale $ρ/2$ is locally 46-sparse. Hence our techniques, which only consider the shortest path cover of scale $ρ/2$, cannot yield a $(2 − ε)$-FPA algorithm for parameter h. The catch is though that the reduction produces graphs which do not have locally sparse shortest path covers for scales significantly larger than $ρ/2$. Hence a $(2 − ε)$-FPA algorithm for parameter h might still exist. However, such an algorithm would have to take larger scales into account than just $ρ/2$, and as mentioned above, it would have to have at least doubly exponential running time in h.

Proving that no $(2 − ε)$-FPA algorithm for parameter k exists for k-CENTER, unless FPT=W[2], is straightforward given the original reduction of Hsu and Nemhauser [20] from the W[2]-hard DOMINATING SET problem. For parameter h, however, we develop some more advanced techniques. For the reduction we show how to construct a graph of low highway dimension given a metric of low *doubling dimension* (see Section 4 for a formal definition), so that distances between vertices are preserved by a $(1 + ε)$ factor. The doubling dimension [18] is a parameter that captures the bounded volume growth of metrics, such as given by Euclidean and Manhattan distances. Since

4

k-CENTER is not $(2 - \varepsilon)$-approximable in L_∞-metrics [14], unless P=NP, and these have constant doubling dimension, we are able to conclude that the hardness translates to graphs of highway dimension $O(\log^2 n)$.

1.2 Generalizations

In addition to k-CENTER, in Section 5 we obtain similar positive results for two generalizations of the problem by appropriately modifying our techniques. For the WEIGHTED k-CENTER problem, the vertices have integer weights and the objective is to choose centers of total weight at most k to cover all vertices with balls of minimum radius. This problem is 3-approximable [19, 29] and no better approximation factor is known. However, we are able to modify our techniques to obtain a 2-FPA algorithm for the combined parameter (k, h).

An alternative way to define the k-CENTER problem is in terms of finding a *star cover* of size k in a metric, where the cost of the solution is the longest of any star edge in the solution. More generally, in their seminal work Hochbaum and Shmoys [19] defined the $(k, \mathcal{F})$-PARTITION problem. Here a family of (unweighted) graphs $\mathcal{F}$ is given and the aim is to partition the vertices of a metric into k sets and connect the vertices of each set by a graph from the family $\mathcal{F}$. The solution cost is measured by the "bottleneck", which is the longest distance between any two vertices of the metric that are connected by an edge in a graph from the family $\mathcal{F}$. The case when $\mathcal{F}$ contains only stars is exactly the k-CENTER problem, given the shortest-path metric as input. The $(k, \mathcal{F})$-PARTITION problem is $2d$-approximable [19], where d is the largest diameter of any graph in $\mathcal{F}$. We show that a 3δ-FPA algorithm for the combined parameter (k, h) exists, where δ is the largest radius of any graph in $\mathcal{F}$. Hence for graph families in which $3\delta < 2d$ this improves on the general algorithm by Hochbaum and Shmoys [19]. This is for example the case when $\mathcal{F}$ contains "stars of paths", i.e., stars for which each edge is replaced by a path of length at most δ. The diameter of such a graph can be 2δ, while the radius is at most δ, and hence $3\delta < 2d = 4\delta$.

1.3 Related work

Given its applicability to various problems in transportation networks, but also in other contexts such as image processing and data-compression, the k-CENTER problem has been extensively studied in the past. We only mention closely related results here, that were not mentioned before. For parameters clique-width and tree-width, Katsikarelis et al. [23] show that k-CENTER is W[1]-hard, but they also give fixed-parameter approximation schemes for each of these parameters. For the tree-depth parameter, they show that the problem is FPT. For unweighted planar and map graphs the k-CENTER problem is FPT [12] for the combined parameter (k, ρ), where ρ is the cost of the optimum solution. Note though that k and ρ are somewhat opposing parameters in the sense that typically if k is small then ρ will be large, and vice versa. A very recent result [24] gives an *efficient* polynomial-time approximation scheme (EPTAS) for k-CENTER on weighted planar graphs, which approximates both the optimum cost ρ and the number of centers k. That is, in time $f(\varepsilon) \cdot n^{O(1)}$ the algorithm computes a $(1 + \varepsilon)$-approximation that uses at most $(1 + \varepsilon)k$ centers, for any $\varepsilon > 0$. Interestingly, this immediately implies a fixed-parameter approximation scheme for parameters k and ε on weighted planar graphs: setting ε to a value smaller than $1/k$ forces the algorithm to compute a solution with at most k centers (since k is an integer), while the cost is within an $(1 + \varepsilon)$-factor of the optimum. Marx and Pilipczuk [27] prove that in planar graphs an optimum k-CENTER solution can be computed in time $n^{O(\sqrt{k})}$. On the other hand, a recent result [15] shows that k-CENTER is W[1]-hard in planar graphs with constant doubling dimension, for the combined parameter (k, h, t), where h is the highway dimension and t the treewidth of the input graph. Thus

this problem remains hard, even when assuming that it abides to all aforementioned models of transportation networks at once. For any L_q metric an $(1 + \varepsilon)$-FPA algorithm for the combined parameter (k, ε, D) can be obtained [4], where D is the dimension of the geometric space. This can also be generalized [15] to an $(1 + \varepsilon)$-FPA algorithm for the combined parameter (k, ε, d), where d is the doubling dimension.

Abraham et al. [1] introduced the highway dimension, and study it in several papers [1, 2, 3]. Their main interest is in explaining the good performance of various shortest-path heuristics assuming low highway dimension. In [2] they show that a locally $O(h \log h)$-sparse shortest path cover can be computed in polynomial time for any scale if the highway dimension of the input graph is h, and each shortest path is unique. Feldmann et al. [16] consider computing approximations for various other problems that naturally arise in transportation networks. They show that quasi-polynomial time approximation schemes can be obtained for problems such as TRAVELLING SALESMAN, STEINER TREE, or FACILITY LOCATION, if the highway dimension is constant. For this however a more restrictive definition of the highway dimension than used here is needed (see [16, Section 9] for more details). The algorithms are obtained by probabilistically embedding a low highway dimension graph into a bounded treewidth graph while introducing arbitrarily small distortions of distances. Known algorithms to compute optimum solutions on low treewidth graphs then imply the approximation schemes. It is interesting to note that this approach does not work for the k-CENTER problem since, in contrast to the above mentioned problems, its objective function is not linear in the edge lengths. As noted before however, a recent result by Becker et al. [10] obtains a fixed-parameter approximation scheme for k-CENTER for combined parameter (h, k, ε) using a deterministic embedding, building on the results in [16]. But again, for this the more restrictive definition of highway dimension also used in [16] is needed. The only other theoretical results on the highway dimension that we are aware of at this point are by Bauer et al. [9] and by Kosowski and Viennot [25]. Bauer et al. [9] show that for any graph G there exist edge lengths such that the highway dimension is $\Omega(\mathrm{pw}(G)/\log n)$, where $\mathrm{pw}(G)$ is the pathwidth of G. Kosowski and Viennot [25] introduce the *skeleton dimension* of a graph and compare it to the highway dimension in the context of shortest path heuristics.

2 k-CENTER and highway dimension versus DOMINATING SET and vertex covers

We begin by observing that the vertices of a low highway dimension graph are highly structured for any scale r: the vertices that are far from any hub of a shortest path cover for scale r are clustered into sets of small diameter and large inter-cluster distance (see Figure 1). A similar observation was already made in [16], where clusters were called *towns*. We need a slightly different definition of clusters than in [16] however, which is why we do not use the same terminology here. For a set $S \subseteq V$ let $\mathrm{dist}_G(u, S) = \min_{v \in S} \mathrm{dist}_G(u, v)$ be the shortest-path distance from u to the closest vertex in S.

Definition 3. Fix $r \in \mathbb{R}^+$ and a shortest path cover $\mathrm{SPC}(r) \subseteq V$ for scale r in a graph $G = (V, E)$. We call an inclusion-wise maximal set $T \subseteq \{v \in V \mid \mathrm{dist}_G(v, \mathrm{SPC}(r)) > r\}$ with $\mathrm{dist}_G(u, w) \leq r$ for all $u, w \in T$ a *cluster*, and we denote the set of all clusters by $\mathcal{T}$. The *non-cluster* vertices are those which are not contained in any cluster of $\mathcal{T}$.

Note that the set $\mathcal{T}$ is specific for the scale r and the hub set $\mathrm{SPC}(r)$. The following lemma summarizes the structure of the clusters and non-cluster vertices. Here we let $\mathrm{dist}_G(S, S') = \min_{v \in S} \mathrm{dist}_G(v, S')$ be the minimum distance between vertices of two sets S and S'.

Lemma 4. *Let $\mathcal{T}$ be the cluster set for a scale r and a shortest path cover $\mathrm{SPC}(r)$. For each non-cluster vertex v, $\mathrm{dist}_G(v, \mathrm{SPC}(r)) \leq r$. The diameter of any cluster $T \in \mathcal{T}$ is at most r, and $\mathrm{dist}_G(T, T') > 2r$ for any distinct pair of clusters $T, T' \in \mathcal{T}$.*

Proof. The first two claims follow immediately from the definition of the clusters. For the third claim let $W = \{v \in V \mid \mathrm{dist}_G(v, \mathrm{SPC}(r)) > r\}$, such that any cluster $T \in \mathcal{T}$ is a subset of W. We first argue that there are no vertices $u, w \in W$ for which $\mathrm{dist}_G(u, w) \in (r, 2r]$. If these existed, by Definition 1 there would be a hub $x \in \mathrm{SPC}(r)$ hitting the shortest path between them. However, this path would have length $\mathrm{dist}_G(u, x) + \mathrm{dist}_G(w, x) > 2r$ since u and w are at distance more than r from $\mathrm{SPC}(r)$, contradicting our assumption that $\mathrm{dist}_G(u, w) \leq 2r$.

As a consequence, for any three vertices $u, v, w \in W$ with $\mathrm{dist}_G(u, v) \leq r$ and $\mathrm{dist}_G(v, w) \leq r$ we have $\mathrm{dist}_G(u, w) \leq \mathrm{dist}_G(u, v) + \mathrm{dist}_G(v, w) \leq 2r$, and since we know that $\mathrm{dist}_G(u, w) \notin (r, 2r]$, this implies that in fact $\mathrm{dist}_G(u, w) \leq r$. Hence the relation of being at distance at most r in W is transitive, and it is obviously also symmetric and reflexive, i.e., it is an equivalence relation on W. Moreover, any two vertices $u, w \in W$ that do not belong to the same equivalence class, i.e. $\mathrm{dist}_G(u, w) > r$, must be at distance more than $2r$, as $\mathrm{dist}_G(u, w) \notin (r, 2r]$. By Definition 3 the clusters are exactly the equivalence classes of W, and so $\mathrm{dist}_G(T, T') > 2r$ for any two distinct clusters $T, T' \in \mathcal{T}$. $\qquad\square$

A *vertex cover* W is a subset of vertices such that every edge is incident to some vertex of W. In particular, if all edges have unit length, then a shortest path cover for scale $r = 1/2$ is a vertex cover. Hence shortest path covers are generalizations of vertex covers. A *dominating set* D is a subset of vertices such that every vertex is adjacent to some vertex of D. In a graph with unit edge lengths, a feasible k-CENTER solution of cost 1 is a dominating set. In this sense the k-CENTER problem is a generalization of the DOMINATING SET problem, for which a dominating set of minimum size needs to be found. The DOMINATING SET problem is $W[2]$-hard [13] for its canonical parameter (i.e., the size of the optimum dominating set), but it is FPT [5] for the parameter given by the *vertex cover number*, which is the size of the smallest vertex cover of a given graph. As the following simple lemma shows, if ρ is the cost of the optimum k-CENTER solution, the number of hubs of the shortest path cover $\mathrm{SPC}(\rho/2)$ is bounded in k and the local sparsity of $\mathrm{SPC}(\rho/2)$. Thus our setting generalizes the DOMINATING SET problem on graphs with bounded vertex cover number. It is interesting to note that in contrast to the DOMINATING SET problem being FPT for the vertex cover number [5], our more general setting is W[1]-hard [15].

Lemma 5. *Let ρ be the optimum cost of the k-CENTER problem in a given instance G. If a shortest path cover $\mathrm{SPC}(\rho/2)$ of G for scale $\rho/2$ is locally s-sparse, then $|\mathrm{SPC}(\rho/2)| \leq ks$.*

Proof. The optimum k-CENTER solution covers the whole graph G with k balls of radius ρ each. By Definition 1 there are at most s hubs of $\mathrm{SPC}(\rho/2)$ in each ball. $\qquad\square$

We are able to exploit this intuition for our algorithm in Section 3. On a high level, our algorithm follows the lines of the following simple procedure to solve DOMINATING SET on graphs with bounded vertex cover number. As a subroutine we will solve an instance of the SET COVER problem, for which a collection $\mathcal{S} \subseteq 2^U$ of subsets of a universe U is given together with a subset $U' \subseteq U$ of the universe.[1] A *set cover* for U' is a collection $\mathcal{S}' \subseteq \mathcal{S}$ of the sets in $\mathcal{S}$ covering U', i.e., $\bigcup_{S \in \mathcal{S}'} S \supseteq U'$. The aim is to compute a minimum-sized set cover for the set system $(U', \mathcal{S})$. Given an input graph $G = (V, E)$ and a vertex cover $W \subseteq V$ of small size (which can, for instance, be an approximation), we perform the following three steps, in each of which we find a respective subset D_i, $i \in \{1, 2, 3\}$, of the optimum dominating set $D \subseteq V$ of G.

---

[1]Usually $U' = U$ but for convenience we define the problem slightly more general here.

1. Guess the subset $D_1 = W \cap D$ of vertices in the vertex cover W that belong to the dominating set D.
2. Since the vertices not in the vertex cover W form an independent set, any vertex of $V \setminus W$, which is not adjacent to a vertex in D_1 must be in D. Thus we can let D_2 consist of all such vertices from $V \setminus W$.
3. By our choice of D_2, if there are any vertices left in V that are not adjacent to $D_1 \cup D_2$, they must be in W. Furthermore these vertices must be adjacent to some vertices in D contained in $V \setminus W$, by our choice of D_1. We can thus solve an instance of SET COVER, where U' is given by the subset of vertices in W that are not adjacent to $D_1 \cup D_2$, and the set system $\mathcal{S}$ is given by the neighbourhoods of vertices in $V \setminus W$ restricted to W. The remaining set $D_3 = D \setminus (D_1 \cup D_2)$ consists of the vertices in $V \setminus W$ whose neighbourhoods form the smallest solution of this SET COVER instance.

For the first step of the above algorithm there are $2^{|W|}$ possible guesses for D_1. For each such guess, the second step can be performed in polynomial time. For the third step we need to solve SET COVER for an instance with a small universe U. This can be done in $2^{|U|} \cdot (|U| + |\mathcal{S}|)^{O(1)}$ time using the algorithm of Fomin et al. [17]. Since in our case $U = W$ and $|\mathcal{S}| \leq |V \setminus W|$, this amounts to a running time of $2^{|W|} \cdot n^{O(1)}$. This SET COVER algorithm is based on dynamic programming. During its execution the smallest set cover for every subset U' of the universe U is computed, and these optimum solutions are stored in a table. Therefore, instead of running an algorithm for SET COVER for each guess of D_1 in the third step above, we may run the algorithm of Fomin et al. [17] only once beforehand: we set the universe to all of W, and the set system will contain all neighbourhood sets of vertices in $V \setminus W$. This way the needed optimum solution for the corresponding subset U' of U can be retrieved in constant time in the third step of our procedure. As we need to retrieve an optimum set cover for every guess of D_1, this improves the overall running time, which is now $2^{|W|} \cdot n^{O(1)}$.

In our k-CENTER algorithm we will use the same method of pre-computing a table containing all optimum SET COVER solutions for subsets of a universe. We summarize the properties of the needed SET COVER algorithm in the following.

Theorem 6 ([11, 17]). *Given a set system $(U, \mathcal{S})$ we can compute a table $\mathbb{T}$, which for any subset $U' \subseteq U$ contains the smallest set cover for $(U', \mathcal{S})$ in the entry $\mathbb{T}[U']$. For any subset $U' \subseteq U$, the optimum set cover for U' can be retrieved in constant time from $\mathbb{T}$, and $\mathbb{T}$ can be computed in $2^{|U|} \cdot (|U| + |\mathcal{S}|)^{O(1)}$ time.*

3 The fixed-parameter approximation algorithm

We begin with a brief high-level description of the algorithm. As observed in Section 2, we can think of solving k-CENTER in a low highway dimension graph as a generalization of solving DOMINATING SET in a graph with bounded vertex cover number. Our algorithm (see Algorithm 1) is driven by this intuition. After guessing the optimum k-CENTER cost ρ and computing SPC($\rho/2$) together with its cluster set $\mathcal{T}$, we will see how the algorithm computes three approximate center sets C_1, C_2, and C_3 (analogous to the three respective sets D_1, D_2, D_3 for DOMINATING SET). For the first set C_1 the algorithm guesses a subset of the hubs of SPC($\rho/2$) that are close to the optimum center set. This can be done in time exponential in k and the local sparsity of the hub set, because there are at most that many hubs for scale $\rho/2$ by Lemma 5. We will observe that by Lemma 4 an optimum center lying in a cluster cannot cover any vertices that are part of another cluster. This makes it easy to determine a second set C_2 of approximate centers, each of which will lie in a cluster that must contain an optimum center. The third set of centers C_3 will consist of cluster vertices that cover the remaining vertices not yet covered by C_1 and C_2. These remaining uncovered vertices

will all be non-cluster vertices, and we find C_3 by solving a SET COVER instance, similar to the third step in our procedure for DOMINATING SET.

More concretely, consider an input graph $G = (V, E)$ with an optimum k-CENTER solution C^* of cost ρ. In line 2 to line 4 of Algorithm 1 we try scales r in increasing order, to guess the correct value for which $r = \rho/2$. For each guessed value of r the algorithm computes a shortest path cover $\mathrm{SPC}(r)$ together with its cluster set $\mathcal{T}$ in line 5. By [2], locally $O(h \log h)$-sparse shortest path covers are computable in polynomial time if the input graph has highway dimension h. In line 1 we therefore set s to the bound of the local sparsity guaranteed in [2] (if locally h-sparse shortest path covers are given by an oracle, we may at this point set $s = h$). In order to keep the running time low, the algorithm checks that the number of hubs is not too large in line 6: since by Lemma 5 we have $|\mathrm{SPC}(\rho/2)| \le ks$, we can dismiss any shortest path cover containing more hubs.

Assume that $r = \rho/2$ was found. In the following, for an index $i \in \{1, 2, 3\}$ we denote by $R_i^* = \bigcup_{v \in C_i^*} B_v(\rho)$ and $R_i = \bigcup_{v \in C_i} B_v(\frac{3}{2}\rho)$ the regions covered by some set of optimum centers $C_i^* \subseteq C^*$ (with balls of radius ρ) and approximate centers $C_i \subseteq V$ (with balls of radius $\frac{3}{2}\rho$), respectively. In line 10 the algorithm guesses a minimum-sized set H of hubs in $\mathrm{SPC}(\rho/2)$, such that the balls of radius $\rho/2$ around hubs in H cover all optimum non-cluster centers. That is, if $C_1^* \subseteq C^*$ denotes the set of optimum non-cluster centers, each of which is at distance at most $\rho/2$ from some hub in $\mathrm{SPC}(\rho/2)$, then $C_1^* \subseteq \bigcup_{v \in H} B_v(\rho/2)$ and $H \subseteq \mathrm{SPC}(\rho/2)$ is a minimum-sized such set. We choose this set of hubs H as the first set of centers C_1 for our approximate solution in line 11. Note that due to the minimality of H we have $|C_1| \le |C_1^*|$. Also $R_1^* \subseteq R_1$ since for any center in C_1^* there is a center (i.e., a hub) at distance at most $\rho/2$ in C_1.

The next step is to compute a set of centers so that all clusters of the cluster set $\mathcal{T}$ of $\mathrm{SPC}(\rho/2)$ are covered. Some of the clusters are already covered by the first set of centers C_1, and thus in this step we want to take care of all remaining uncovered clusters, i.e., those contained in $\mathcal{U} = \{T \in \mathcal{T} \mid T \setminus R_1 \ne \emptyset\}$. By the definition of C_1^*, any remaining optimum center in $C^* \setminus C_1^*$ must lie in a cluster. Furthermore, the distance between clusters of $\mathrm{SPC}(\rho/2)$ is more than ρ by Lemma 4, so that a center of $C^* \setminus C_1^*$ in a cluster T cannot cover any vertices of another cluster $T' \ne T$. Hence if we guessed H correctly, we can be sure that each cluster $T \in \mathcal{U}$ must contain a center of $C^* \setminus C_1^*$. For each (remaining) cluster $T \in \mathcal{U}$ we thus pick an arbitrary vertex $v \in T$ in line 16 and declare it a center of the second set C_2 for our approximate solution. Hence if the optimum set of centers for $\mathcal{U}$ is $C_2^* = \{v \in C^* \mid \exists T \in \mathcal{U} : v \in T\}$, we have $|C_2| \le |C_2^*|$ (if some cluster of $\mathcal{U}$ contains more than one optimum center in order to cover different parts of the non-cluster vertices, C_2^* may be larger than C_2). Moreover, since the diameter of each cluster is at most $\rho/2$ by Lemma 4, we get $R_2^* \subseteq R_2$.

At this time we know that all clusters in $\mathcal{T}$ are covered by the region $R_1 \cup R_2$. Hence if any uncovered vertices remain in $V \setminus (R_1 \cup R_2)$ for our current approximate solution, they must be non-cluster vertices. Just as C_2^*, by our definition of C_1^*, every remaining optimum center in $C_3^* = C^* \setminus (C_1^* \cup C_2^*)$ lies in some cluster. Since $R_1^* \subseteq R_1$ and $R_2^* \subseteq R_2$, any remaining uncovered vertex of $V \setminus (R_1 \cup R_2)$ must be in the region R_3^* covered by centers in C_3^*. Next we show how to compute a set C_3 such that the region R_3 includes all remaining vertices of the graph and $|C_3| \le |C_3^*|$. Note that the latter means that the number of centers in $C_1 \cup C_2 \cup C_3$ is at most k, since C_1^*, C_2^*, and C_3^* are disjoint.

To control the size of C_3 we will compute the smallest number of centers that cover parts of R_3^* with balls of radius ρ. In particular, in line 18 we guess the set of hubs $H' \subseteq \mathrm{SPC}(\rho/2) \setminus H$ that lie in the region R_3^* (note that we exclude hubs of H from this set). We then compute a center set C_3 of minimum size such that $H' \subseteq \bigcup_{v \in C_3} B_v(\rho)$. For this we reduce the problem of computing centers covering H' to the SET COVER problem with fixed universe size, as shown in line 7 to line 9. This reduction is performed before entering the loops guessing H and H' to optimize the running time.

The universe U of the SET COVER instance consists of all hubs in the shortest path cover $\textsc{spc}(r)$, while the set system $\mathcal{S}$ of the instance is obtained by restricting the balls $B_v(\rho)$ of radius ρ around cluster vertices v to the hubs. By Theorem 6 there is an algorithm that computes the optimum SET COVER solution for every subset of the universe. This algorithm is called in line 9 of Algorithm 1 to fill a lookup table $\mathbb{T}$ with these optima. We can thus retrieve the optimum SET COVER solution for the subset $H' \subseteq \textsc{spc}(r)$ in line 19, and let C_3 contain each cluster vertex v for which the set of hubs contained in the ball $B_v(\rho)$ is part of the optimum solution covering H', which is stored in the entry $\mathbb{T}[H']$ of the table. As the next lemma shows, we obtain the required properties for C_3.

Lemma 7. *Assume the algorithm guessed the correct scale $r = \rho/2$ and the correct sets H and H'. The set $C_3 = \{v \in \bigcup_{T \in \mathcal{T}} T \mid B_v(\rho) \cap \textsc{spc}(r) \in \mathbb{T}[H']\}$ is of size at most $|C_3^*|$ and $H' \subseteq \bigcup_{v \in C_3} B_v(\rho)$.*

Proof. The second property clearly follows since the sets $B_v(\rho) \cap \textsc{spc}(r)$ in $\mathbb{T}[H']$ form a set cover for H', such that every hub in H' is at distance at most ρ from some $v \in C_3$. To see that $|C_3| \leq |C_3^*|$, it suffices to show that the vertices in C_3^* correspond to a feasible SET COVER solution for H'. If H' was guessed correctly, this set contains only hubs in the region R_3^*. As R_3^* is covered by balls of radius ρ around the centers in C_3^*, the union $\bigcup_{v \in C_3^*} B_v(\rho) \cap \textsc{spc}(r)$ contains H'. Moreover, these sets $B_v(\rho) \cap \textsc{spc}(r)$ are contained in the set system $\mathcal{S}$, since all centers of C_3^* are contained in clusters by definition of C_1^*. Thus the sets $B_v(\rho) \cap \textsc{spc}(r)$ form a set cover for H' in the instance $(\textsc{spc}(r), \mathcal{S})$. $\qquad\square$

It remains to show that the three computed center sets C_1, C_2, and C_3 cover all vertices of G, which we do in the following lemma. In particular, the union $C_1 \cup C_2 \cup C_3$ will pass the feasibility test in line 22 of the algorithm.

Lemma 8. *Assume the algorithm guessed the correct scale $r = \rho/2$ and the correct sets H and H'. The approximate center sets C_1, C_2, and C_3 cover all vertices of G, i.e., $R_1 \cup R_2 \cup R_3 = V$.*

Proof. The proof is by contradiction: assume there is a $v \in V \setminus (R_1 \cup R_2 \cup R_3)$ that is not covered by the computed approximate center sets. The idea is to identify a hub $y \in \textsc{spc}(\rho/2)$ on the shortest path between v and an optimum center $w \in C^*$ covering v. We will show that this hub y must however be in H' and therefore v is in fact in R_3, since v also turns out to be close to y.

To show the existence of y, we begin by arguing that the closest hub $x \in \textsc{spc}(\rho/2)$ to v is neither in H nor in H'. We know that each cluster of $\mathcal{T}$ is in $R_1 \cup R_2$, so that $v \notin R_1 \cup R_2$ must be a non-cluster vertex. Thus by Lemma 4, $\text{dist}_G(v, x) \leq \rho/2$. The region R_1 in particular contains all vertices that are at distance at most $\rho/2$ from any hub in $H = C_1$. Since $v \notin R_1$ and $\text{dist}_G(v, x) \leq \rho/2$, this means that $x \notin H$. From $v \notin R_3$ we can also conclude that $x \notin H'$ as follows. By Lemma 7, C_3 covers all hubs of H' with balls of radius ρ. Hence if $x \in H'$ then v would be at distance at most $\frac{3}{2}\rho$ from a center of C_3, i.e., $v \in R_3$.

From $x \notin H \cup H'$ we can conclude the existence of y as follows. Consider an optimum center $w \in C^*$ that covers v, i.e., $v \in B_w(\rho)$. Recall that $R_1^* \subseteq R_1$ and $R_2^* \subseteq R_2$. Since $v \notin R_1 \cup R_2$, this means that w is neither in C_1^* nor in C_2^* so that $w \in C_3^*$. By definition of H', any hub at distance at most ρ from a center in C_3^* is in H', unless it is in H. Hence, as $x \notin H \cup H'$, the distance between x and w must be more than ρ. Since $\text{dist}_G(v, x) \leq \rho/2$, we get $\text{dist}_G(v, w) > \rho/2$. We also know that $\text{dist}_G(v, w) \leq \rho$, because w covers v. Hence the shortest path cover $\textsc{spc}(\rho/2)$ must contain a hub y that lies on the shortest path between v and w. In particular, $\text{dist}_G(v, y) \leq \rho$ and $\text{dist}_G(y, w) \leq \rho$. Analogous to the argument used for x above, R_1 in particular contains all vertices at distance at most ρ from H, so that $y \notin H$ since $v \notin R_1$. However, then the distance bound for y and w yields $y \in H'$, as $w \in C_3^*$.

Since C_1^* contains all non-cluster centers but $w \notin C_1^*$, by Lemma 4 we get $\text{dist}_G(y, w) > \rho/2$, which implies $\text{dist}_G(v, y) < \rho/2$. But then v is contained in the ball $B_y(\rho/2)$, which we know is part

of the third region R_3 since $y \in H'$. This contradicts the assumption that v was not covered by the approximate center set. $\qquad\square$

Note that the proof of Lemma 8 does not imply that $R_3^* \subseteq R_3$, as was the case for R_1 and R_2. It suffices though to establish the correctness of the algorithm. Finally, we conclude the proof of Theorem 2 by analysing the runtime of the algorithm.

Proof of Theorem 2. By Lemma 4 and Lemma 7, if Algorithm 1 correctly guesses the cost ρ and the two hub sets H and H', then $|C_1 \cup C_2 \cup C_3| \leq k$ and $R_1 \cup R_2 \cup R_3 = V$. By Lemma 5,

---

Algorithm 1: FPA algorithm for k-CENTER in low highway dimension graphs

Input: Graph $G = (V, E)$ of highway dimension h with optimum k-CENTER cost ρ
Output: k-CENTER set C of cost at most $\frac{3}{2}\rho$

1   $s \leftarrow O(h \log h)$ // local sparsity of efficiently computable shortest path cover

2   $\mathbb{A} \leftarrow \mathtt{sort}(\{\mathrm{dist}_G(u, v) \mid u, v \in V\})$ // sort distances and store them in array $\mathbb{A}$
3   **for** $i \leftarrow 0$ **to** $\binom{n}{2} - 1$ **do** // consider distances in increasing order
4      $r \leftarrow \mathbb{A}[i]/2$ // guess $r = \rho/2$

5   Compute locally s-sparse $\mathrm{SPC}(r)$ with cluster set $\mathcal{T}$
6   **if** $|\mathrm{SPC}(r)| > ks$ **then continue** // too many hubs means $r \neq \rho/2$

     // prepare the SET COVER lookup table
7      $V(\mathcal{T}) \leftarrow \bigcup_{T \in \mathcal{T}} T$
8      $\mathcal{S} \leftarrow \bigcup_{v \in V(\mathcal{T})} \{B_v(\rho) \cap \mathrm{SPC}(r)\}$ // the set system is given by hubs in balls of
        radius ρ around cluster vertices
9      $\mathbb{T} \leftarrow \mathtt{SetCoverDP}(\mathrm{SPC}(r), \mathcal{S})$ // lookup table $\mathbb{T}$ contains an optimum set cover
        for every subset of the universe $\mathrm{SPC}(r)$

     // guess minimum-sized set of hubs covering non-cluster centers
10   **foreach** $H \subseteq \mathrm{SPC}(r)$ **do**
11        $C_1 \leftarrow H$ // these hubs form the 1st set of centers

       // cover all clusters not covered by balls around H:
12        $R_1 \leftarrow \bigcup_{v \in C_1} B_v(3r)$ // the region covered so far
13        $\mathcal{U} \leftarrow \{T \in \mathcal{T} \mid T \setminus R_1 \neq \emptyset\}$ // the clusters that still need to be covered
14        $C_2 \leftarrow \emptyset$
15   **foreach** $T \in \mathcal{U}$ **do**
16           $v \in T$ // select arbitrary vertex in T
17           $C_2 \leftarrow C_2 \cup \{v\}$ // the 2nd set of centers

       // cover rest of non-cluster vertices by reducing to SET COVER:
18   **foreach** $H' \subseteq \mathrm{SPC}(r) \setminus H$ **do** // guess hubs covered by centers in clusters
19           $C_3 \leftarrow \{v \in V(\mathcal{T}) \mid B_v(\rho) \cap \mathrm{SPC}(r) \in \mathbb{T}[H']\}$ // the 3rd set of centers is
            given by cluster vertices whose balls of radius ρ cover H'

          // check whether the solution is feasible:
20           $C \leftarrow C_1 \cup C_2 \cup C_3$
21           $R \leftarrow \bigcup_{v \in C} B_v(3r)$ // the covered region
22   **if** $|C| \leq k$ **and** $R = V$ **then return** C // a feasible solution was found

$|\text{SPC}(\rho/2)| \le ks$ so that the correct value for r will not be skipped in line 6. Hence by trying all possible values for ρ in increasing order, Algorithm 1 will terminate with a feasible solution that covers all vertices with balls of radius $\frac{3}{2}\rho$, due to line 22. To prove Theorem 2 it remains to bound the running time.

There are at most $\binom{n}{2}$ possible values for ρ that need to be tried by the outermost loop, one for every pair of vertices. Hence the only steps of Algorithm 1 that incur exponential running times are when guessing H and H' and when filling the table $\mathbb{T}$ of the dynamic program for the SET COVER problem. These steps are only performed for shortest path covers of size at most ks due to line 6. Since we explicitly exclude the hubs in H when choosing H', each hub of a shortest path cover can either be in H, in H', or in none of them when trying all possibilities. Hence this gives 3^{ks} possible outcomes. Filling the table $\mathbb{T}$ takes $O(2^{ks} \cdot n^{O(1)})$ time according to Theorem 6, while retrieving an optimum solution for H' in line 19 can be done in constant time. Thus the total running time to compute a 3/2-approximation is $O(3^{ks} \cdot n^{O(1)})$. If the input graph has highway dimension h, Abraham et al. [2] show how to compute $O(\log h)$-approximations of shortest path covers in polynomial time if shortest paths have unique lengths. The latter can be assumed by slightly perturbing the edge lengths in such a way that any 3/2-approximation in the perturbed instance also is a 3/2-approximation in the original instance. Therefore we can set $s = O(h \log h)$ during the execution of our algorithm. If there is an oracle that gives locally h-sparse shortest path covers for each scale, then we can set $s = h$ instead. Thus the claimed running times follow. $\square$

4 Hardness results

We begin by observing that the original reduction of Hsu and Nemhauser [20] for k-CENTER also implies that there are no $(2 - \varepsilon)$-FPA algorithms.

Theorem 9. *It is W[2]-hard for parameter k to compute a $(2 - \varepsilon)$-approximation to the k-CENTER problem for any $\varepsilon > 0$.*

Proof (cf. [20, 29]). The reduction is from the DOMINATING SET problem, which is W[2]-hard [13] for the standard parameter, i.e., the size of the smallest dominating set D of the input graph G. The reduction simply introduces unit lengths for each edge of G, guesses the size of D, and sets $k = |D|$. Any feasible center set of cost 1 corresponds to a dominating set, and vice versa. On the other hand, a center set has cost at least 2 if and only if it is not a dominating set. Hence if the size of D is guessed in increasing order starting from 1, k must be equal to $|D|$ the first time a $(2 - \varepsilon)$-approximation of cost 1 is obtained by an algorithm for k-CENTER. By guessing the size of D in increasing order, this would result in an $f(|D|) \cdot n^{O(1)}$ time algorithm to compute the optimum dominating set if there was a $(2 - \varepsilon)$-FPA algorithm for parameter k for k-CENTER. $\square$

We now turn to proving that $(2 - \varepsilon)$-approximations are hard to compute on graphs with low highway dimension. For this we introduce a general reduction from low doubling metrics to low highway dimension graphs in the next lemma. A metric (X, dist_X) has *doubling dimension* d if for every $r \in \mathbb{R}^+$, each set $S \subseteq X$ of diameter $2r$ is the union of at most 2^d sets of diameter r. The *aspect ratio* α of a metric (X, dist_X) is the maximum distance between any two vertices of X divided by the minimum distance, i.e., $\alpha = \max\{\frac{\text{dist}_X(s,t)}{\text{dist}_X(u,v)} \mid s, t, u, v \in X \wedge u \ne v\}$.

Lemma 10. *Given any metric (X, dist_X) with constant doubling dimension d and aspect ratio α, for any $0 < \varepsilon < 1$ there is a graph $G = (X, E)$ of highway dimension $O((\log(\alpha)/\varepsilon)^d)$ on the same vertex set such that for all $u, v \in X$, $\text{dist}_X(u, v) \le \text{dist}_G(u, v) \le (1 + \varepsilon) \text{dist}_X(u, v)$. Furthermore, G can be computed in polynomial time from the metric.*

12

Proof. First off, by scaling we may assume w.l.o.g. that the minimum distance of the given metric is $\frac{2}{1+\varepsilon}$. In particular this means that the maximum distance is $\frac{2\alpha}{1+\varepsilon}$. A fundamental property [18] of low doubling dimension metrics is that for any set of points $Y \subseteq X$ with aspect ratio α', the number of points $|Y|$ can be at most $2^{d\lceil \log_2 \alpha'\rceil}$. The proof of this property is a simple recursive application of the doubling dimension definition. For each scale 2^i where $i \in \{0, 1, \ldots, \lceil \log_2 \alpha \rceil\}$ we will identify a sparse set Y_i, which in any ball of radius 2^{i+1} has aspect ratio $O(\log(\alpha)/\varepsilon)$. The idea is to use the vertices of Y_i as hubs in a shortest path cover for scale 2^i, which then are locally sparse in any such ball. We will make sure that there is an index i with a hub set Y_i for any possible distance between vertex pairs in the resulting graph G. We need to make sure though that the shortest path for any pair of vertices passes through a corresponding hub of some Y_i. We achieve this by adding edges between the hubs in Y_i, which act as shortcuts. That is, the edges of G will be slightly longer than the distances in the metric given by dist_X, and we will make the distances shorter with increasing scales in order to guarantee that the shortest paths pass through corresponding hubs.

More concretely, consider any set $Z \subseteq X$ of vertices. A subset $Y \subseteq Z$ is a *ρ-cover* of Z if for every $v \in Z$ there is a $u \in Y$ such that $\text{dist}_X(u, v) \leq \rho$, and Y is a *ρ-packing* of Z if $\text{dist}_X(u, v) > \rho$ for all distinct $u, v \in Y$. A *ρ-net* of Z is a set $Y \subseteq Z$ that is a ρ-cover and a ρ-packing of Z. It is easy to see that such a net can be computed greedily in $O(n^2)$ time. We will use sets Y_i that form a *hierarchy* $Y_i \subseteq Y_{i-1}$ of nets as hubs. In particular, $Y_0 = X$ and Y_i is a $\frac{\varepsilon 2^{i-3}}{(1+\varepsilon)^2 L}$-net of Y_{i-1} for each $i \geq 1$, where $L = \lceil \log_2 \alpha \rceil$ is the index of the largest scale. Note that due to the triangle inequality of the metric, each Y_i is a $2\frac{\varepsilon 2^{i-3}}{(1+\varepsilon)^2 L}$-cover of X.

In G, for each i we connect two vertices $u, v \in Y_i$ by an edge uv of length $(1+\varepsilon(1-i/L))\text{dist}_X(u, v)$. If a vertex pair is contained in several sets Y_i of different scales, we only add the shortest edge according to the above rule, i.e., the edge for the largest index i. Hence the distance in G between any $u, v \in Y_i$ is at most $(1+\varepsilon(1-i/L))\text{dist}_X(u, v)$. In particular, $\text{dist}_G(u, v) \leq (1+\varepsilon)\text{dist}_X(u, v)$ for any $u, v \in X$ since $X = Y_0$. Note also that $1 + \varepsilon(1-i/L) \geq 1$ and hence $\text{dist}_X(u, v) \leq \text{dist}_G(u, v)$.

To bound the highway dimension of G, consider any pair $u, v \in X$, and let $i \in \{0, 1, \ldots, L\}$ be such that $\text{dist}_G(u, v) \in (2^i, 2^{i+1}]$. Recall that the minimum distance according to dist_X is $\frac{2}{1+\varepsilon} > 1$ (as $\varepsilon < 1$), while the maximum distance is $\frac{2\alpha}{1+\varepsilon}$. Accordingly, in G all distances lie in $(1, 2\alpha]$ so the index i exists. We show that the shortest path between u and v passes through a hub of Y_i. We do this by upper bounding $\text{dist}_G(u, v)$ in terms of $\text{dist}_X(u, v)$ using a path that contains vertices of Y_i. Then we lower bound the length of any path that does not pass through Y_i and show that it is longer than the shortest path.

Let $x \in Y_i$ be the closest hub to u and let $y \in Y_i$ be the closest hub to v. We begin by determining some distance bounds for these vertices. Since Y_i is a $2\frac{\varepsilon 2^{i-3}}{(1+\varepsilon)^2 L}$-cover of X in the metric according to dist_X, the distances in G from u to x and from v to y are at most $2(1 + \varepsilon)\frac{\varepsilon 2^{i-3}}{(1+\varepsilon)^2 L} = \frac{\varepsilon 2^{i-2}}{(1+\varepsilon)L}$ each. It also means that $\text{dist}_X(x, y) \leq \text{dist}_X(u, v) + 2 \cdot \frac{\varepsilon 2^{i-2}}{(1+\varepsilon)^2 L}$, since we can get from x to y through u and v in the metric. We know that $\text{dist}_G(u, v) > 2^i$ and thus we have $\frac{2^i}{1+\varepsilon} < \text{dist}_X(u, v)$. Using

these bounds we get

$$\text{dist}_G(u,v) \le \text{dist}_G(u,x) + \text{dist}_G(x,y) + \text{dist}_G(y,v)$$
$$\le \left[1 + \varepsilon\left(1 - \frac{i}{L}\right)\right]\text{dist}_X(x,y) + 2 \cdot \frac{\varepsilon 2^{i-2}}{(1+\varepsilon)L}$$
$$< \left[1 + \varepsilon\left(1 - \frac{i}{L}\right)\right]\left(\text{dist}_X(u,v) + 2 \cdot \frac{\varepsilon 2^{i-2}}{(1+\varepsilon)^2 L}\right) + \frac{\varepsilon}{2L}\text{dist}_X(u,v)$$
$$< \left[1 + \varepsilon\left(1 - \frac{i}{L}\right) + \left(1 + \varepsilon\left(1 - \frac{i}{L}\right)\right)\frac{\varepsilon}{2(1+\varepsilon)L} + \frac{\varepsilon}{2L}\right]\text{dist}_X(u,v)$$
$$\le \left[1 + \varepsilon\left(1 - \frac{i}{L}\right) + (1+\varepsilon)\frac{\varepsilon}{2(1+\varepsilon)L} + \frac{\varepsilon}{2L}\right]\text{dist}_X(u,v)$$
$$= \left[1 + \varepsilon\left(1 - \frac{i-1}{L}\right)\right]\text{dist}_X(u,v).$$

We now show that every path P that does not use any hub of Y_i is longer than $\text{dist}_G(u,v)$. Since the hub sets of different scales form a hierarchy, any hub of scale 2^j with $j > i$ is also a hub for scale 2^i. Hence if P does not pass through any hub of Y_i, it also cannot contain any vertex of Y_j where $j > i$. Thus, if $P = (w_0, \ldots, w_l)$ where $w_0 = u$ and $w_l = v$, any edge $w_j w_{j+1}$ on P will be of length at least $(1 + \varepsilon(1 - (i-1)/L))\text{dist}_X(w_j, w_{j+1})$. The sum $\sum_{j=0}^{l-1}\text{dist}_X(w_j, w_{j+1})$ of all the distances in the metric over the path P is an upper bound on $\text{dist}_X(u,v)$, and thus the length of P is at least $(1 + \varepsilon(1 - (i-1)/L))\text{dist}_X(u,v)$. Since the distance $\text{dist}_G(u,v)$ is strictly smaller than this bound by the above calculations, the shortest path between u and v in G passes through some hub of Y_i.

To bound the highway dimension, for any $r > 0$ we still need to bound the number of hubs that hit shortest paths of length in $(r, 2r]$ in a ball B of radius $2r$ in G. Since our hub sets form a hierarchy, we may consider all shortest paths longer than r: if i is the index such that $r \in (2^i, 2^{i+1}]$, all shortest paths of length more than 2^i are hit by hubs of Y_i because $Y_j \subseteq Y_i$ for all $j > i$. In G the ball B has a diameter of at most $4r$. Measured in the metric according to dist_X the set of vertices in B also has a diameter of at most $4r \le 2^{i+3}$, since $\text{dist}_X(u,v) \le \text{dist}_G(u,v)$ for any vertices $u, v \in X$. Because Y_i is a $\frac{\varepsilon 2^{i-3}}{(1+\varepsilon)^2 L}$-packing in the metric, the aspect ratio of $Y_i \cap B$ is $\alpha' \le 64(1+\varepsilon)^2 L/\varepsilon$. By the fundamental property of low doubling metrics [18] mentioned above, there are at most $(128(1+\varepsilon)^2 L/\varepsilon)^d$ hubs in $Y_i \cap B$, which concludes the proof. $\qquad\square$

Feder and Greene [14] show that, for any $\varepsilon > 0$, it is NP-hard to compute a $(2-\varepsilon)$-approximation for the k-Center problem in two-dimensional L_∞ metrics. In particular, the metric is induced by a grid graph with unit edge lengths, so that the aspect ratio is at most n. The doubling dimension of any such metric is 2, since a vertex set of diameter $2r$ (contained in a "square" of side-length $2r$) can be covered by 4 vertex sets of diameter r (contained in "squares" of side-length r). By the reduction given in Lemma 10 we can thus obtain graphs of highway dimension $O(\log^2 n)$ for which computing $(2 - \varepsilon)$-approximations to k-Center is NP-hard. The challenge remains is to push the highway dimension bound of this inapproximability result down to a constant. This would mean that no $(2 - \varepsilon)$-FPA algorithm for k-Center exists if the parameter is the highway dimension h, unless P=NP. However, we can still argue that assuming the exponential time hypothesis (ETH) [21, 22], any $(2-\varepsilon)$-FPA algorithm for parameter h must have doubly exponential running time. In particular, the above hardness result implies a polynomial-time reduction from SAT to k-Center on graphs of highway dimension $O(\log^2 n)$. That is, given a SAT formula of size N, the reduction will produce a graph with $n = N^{O(1)}$ vertices and highway dimension

$h = O(\log^2 n) = O(\log^2 N)$. Thus an algorithm computing a $(2 - \varepsilon)$-approximation to k-CENTER in time $2^{2^{o(\sqrt{h})}} \cdot n^{O(1)}$ would be able to decide SAT in time $2^{o(N)} \cdot N^{O(1)}$. However, this would contradict ETH. Thus if a $(2 - \varepsilon)$-approximation algorithm for k-CENTER with parameter h exists, it is fair to assume that its running time dependence on h must be extremely large. To summarize we obtain the following lower bounds.

Corollary 11. *For any constant $\varepsilon > 0$ it is NP-hard to compute a $(2 - \varepsilon)$-approximation for the k-CENTER problem on graphs of highway dimension $O(\log^2 n)$. Moreover, there is no $(2 - \varepsilon)$-FPA for k-CENTER parameterized by the highway dimension h with runtime $2^{2^{o(\sqrt{h})}} \cdot n^{O(1)}$, unless ETH fails.*

The following lemma gives further evidence that obtaining a $(2-\varepsilon)$-FPA algorithm for parameter h is hard. As argued below, it excludes the existence of such algorithms that only use shortest path covers of constant scales.

Lemma 12. *For any $\varepsilon > 0$ it is NP-hard to compute a $(2 - \varepsilon)$-approximation for the k-CENTER problem on graphs for which on any scale $r > 0$ there is a locally $(3 \cdot 2^{2r} - 2)$-sparse shortest path cover SPC(r). Moreover, this is true for instances where the optimum cost ρ is at most 4.*

Proof. The reduction is similar to the one used for Theorem 9, but reduces from the NP-hard DOMINATING SET problem on cubic graphs [6]. To obtain an instance of k-CENTER, again we simply introduce unit edge lengths, guess the size of the minimum dominating set D, and set $k = |D|$. In contrast to the reduction of Theorem 9 however, we will guess the size of D in decreasing order starting from n. As before, any feasible center set of cost 1 corresponds to a dominating set, and vice versa, while on the other hand, a center set has cost at least 2 if and only if it is not a dominating set. Hence whenever k is at least $|D|$ a $(2 - \varepsilon)$-approximation for k-CENTER must have cost $\rho = 1$, and the cost is at least 2 for smaller k. Therefore guessing the size of D in decreasing order, k is equal to $|D|$ the last time a $(2 - \varepsilon)$-approximation of cost 1 is computed by an algorithm for k-CENTER.

Consider the value $k = |D| - 1$, i.e., the iteration at which we realize the size of D. If the number of connected components of the input graph exceeds k, we know that there cannot be a dominating set of size k, and we can dismiss this value as a guess for the size of D right away. Otherwise, there is a connected component with at least two vertices of D, since $|D| = k + 1$. It is easy to see that removing one of these two vertices results in a center set of size k with cost at most 4. Hence we only need to call the $(2 - \varepsilon)$-approximation algorithm for k-CENTER on instances where the optimum cost is $\rho \leq 4$.

It is easy to see that any ball with radius $2r$ around a vertex v contains at most $3(\sum_{i=0}^{2r-1} 2^i) + 1 = 3 \cdot 2^{2r} - 2$ vertices, due to the bound on the maximum degree. Hence any set of hubs is locally $(3 \cdot 2^{2r} - 2)$-sparse, which concludes the proof. □

Consider a $(2 - \varepsilon)$-FPA algorithm for k-CENTER, which only takes shortest path covers of constant scales into account, where the parameter is their sparseness. That is, the algorithm computes a $(2 - \varepsilon)$-approximation using hub sets SPC(r) only for values $r \leq R$ for some $R \in O(1)$, and the parameter is a value s such that SPC(r) is locally s-sparse for every $r \leq R$. By Lemma 12 such an algorithm would imply that P=NP. Moreover this is true even if $R \in O(\rho)$. Hence if it is possible to beat the inapproximability barrier of 2 using the local sparseness as a parameter, then such an algorithm would have to take large (non-constant) scales into account. Note that the running time of our 3/2-FPA algorithm can in fact be bounded in terms of the local sparseness of SPC$(\rho/2)$ instead of the highway dimension. The instances produced by the reduction of Lemma 12

15

have shortest path covers that are locally 46-sparse on scale $r = \rho/2 \leq 2$. Thus we obtain the following corollary, which is a matching hardness lower bound to our algorithm.

Corollary 13. *For any $\varepsilon > 0$ it is NP-hard to compute a $(2 - \varepsilon)$-approximation for the k-CENTER problem on graphs for which on scale $r = \rho/2 \leq 2$ there is a locally 46-sparse shortest path cover $\mathrm{SPC}(r)$, where ρ is the optimum k-CENTER cost.*

From this corollary and Theorem 9, we can conclude that our algorithm necessarily needs to combine the parameter h with k in order to achieve its approximation guarantee of $3/2$.

5   Generalizations of the k-CENTER problem

The WEIGHTED k-CENTER **problem** is defined by giving each vertex $v \in V$ an integer weight $w(v) \in \mathbb{N}$. The aim is to find a set $C \subseteq V$ of centers such that their total weight is at most k, i.e., $\sum_{v \in C} w(v) \leq k$, and the maximum distance of any vertex to its closest center is minimized. Hochbaum and Shmoys [19] gave a 3-approximation to the problem, and no better approximation factor is known. However, Algorithm 1 can be modified to obtain a 2-FPA algorithm for WEIGHTED k-CENTER for parameters k and h in graphs of highway dimension h.

For this, Algorithm 1 will again guess $r = \rho/2$ in line 2 to line 4, where ρ is the cost of an optimum solution. The three center sets C_1, C_2, C_3 will be chosen more carefully respecting the weights. In particular, instead of setting $C_1 = H$ in line 11, for each hub $x \in H$ we pick a cheapest vertex in the ball $B_x(r)$ around x to be a center of C_1, i.e., we pick a vertex from $\arg\min\{w(u) \mid u \in B_x(r)\}$. If H was guessed correctly so that each non-cluster center $u \in C_1^*$ of the optimum solution C^* has a hub of H at distance at most r, then there is also a center v in C_1 at distance at most $2r$ from u. Hence a ball of radius $4r$ around v will contain the ball of radius $2r$ around u, i.e., $R_1^* \subseteq \bigcup_{v \in C_1} B_v(4r)$. Furthermore, $w(v) \leq w(u)$ and hence the total weight of C_1 is at most that of C_1^*.

In line 16 of Algorithm 1, instead of picking an arbitrary vertex of the cluster T, we will pick a vertex of T with minimum weight. Since the choice of a vertex in T was arbitrary before, we still have $R_2^* \subseteq R_2$. Additionally the total weight of C_2 is at most that of C_2^* since each cluster of $\mathcal{U}$ contains a center of C_2^* if H was guessed correctly. To compute C_3 we will solve the WEIGHTED SET COVER problem in line 9, where the weight of each set $B_v(\rho) \cap \mathrm{SPC}(r)$ equals $w(v)$. This can easily be done by adapting the dynamic program of Fomin et al. [17] to respect weights of sets (cf. [11]). Hence the weight of the resulting center set C_3 is at most that of C_3^*, and balls of radius $3r$ around centers in C_3 still cover all remaining vertices, by the same arguments as in the proof of Lemma 8.

In conclusion, the set of centers $C = C_1 \cup C_2 \cup C_3$ computed by the modified algorithm has a total weight at most that of C^*, and balls of radius $4r$ around the centers in C cover all vertices. Since the weights are integers, there are at most ks hubs for scale $\rho/2$ if $\mathrm{SPC}(\rho/2)$ is locally s-sparse (cf. Lemma 5). Therefore we obtain a 2-FPA algorithm for the combined parameter (k, h).

For the $(k, \mathcal{F})$-PARTITION **problem** a family $\mathcal{F}$ of unweighted graphs is given, such that for any $n \in \mathbb{N}$ there is a graph in $\mathcal{F}$ with exactly n vertices. Given an input metric (X, dist_X) and a value $c \in \mathbb{R}^+$, the *bottleneck graph* $H_X(c)$ on vertex set X has an edge for every pair of vertices $u, v \in X$ with $\mathrm{dist}_X(u, v) \leq c$. For the $(k, \mathcal{F})$-PARTITION problem the minimum cost c needs to be found such that X can be partitioned into k sets $X_1, \ldots, X_k$, and there is a spanning subgraph $G \in \mathcal{F}$ in $H_X(c)$ on the vertex set X_j for each $j \in \{1, \ldots, k\}$.

Note that if $\mathcal{F} = \{K_{1,i}\}_{i \geq 0}$, i.e., each graph in the family is a star, we have the k-CENTER problem, and if $\mathcal{F} = \{K_i\}_{i \geq 1}$, i.e., each graph in the family is a clique, we have the so-called k-CLUSTERING problem. The *eccentricity* of a vertex v is the maximum distance from v to any other vertex in terms of number of edges (i.e., measured by the *hop-distance*). The *diameter* of an

16

unweighted graph G is defined as the maximum eccentricity of any vertex in G. If the diameter of each $G \in \mathcal{F}$ is at most d, then a $2d$-approximation can be obtained for the $(k, \mathcal{F})$-PARTITION problem [19].

Let the *radius* of a graph G be the minimum eccentricity of any vertex in G. For shortest-path metrics induced by graphs of highway dimension h, we can obtain a 3δ-FPA algorithm for the combined parameter (k, h) for the $(k, \mathcal{F})$-PARTITION problem, if every graph in the family $\mathcal{F}$ has radius at most δ. Hence for graph families $\mathcal{F}$ for which $3\delta < 2d$, this improves on the $2d$-approximation by Hochbaum and Shmoys [19]. This is, for example, the case when $\mathcal{F}$ contains "stars of paths", i.e., stars for which each edge is replaced by a path of length at most δ. The diameter of such a graph can be 2δ, while the radius is at most δ, and hence $3\delta < 2d = 4\delta$.

To obtain our algorithm we reduce the $(k, \mathcal{F})$-PARTITION problem to k-CENTER. Note that if there is an optimum solution to $(k, \mathcal{F})$-PARTITION with cost ρ, then there must be a solution of cost $\delta\rho$ for k-CENTER: for each X_j^* of the optimum partition for $(k, \mathcal{F})$-PARTITION, place a center at a vertex v of X_j^* that minimizes the eccentricity in the graph $G \in \mathcal{F}$ spanning X_j^*. Since every edge of G has length at most ρ, the ball $B_v(\delta\rho)$ will contain G. Computing a $3/2$-approximation to k-CENTER using Algorithm 1, we obtain a set C of k centers such that the closest center to any vertex is at distance at most $3\delta\rho/2$. For each center $v_j \in C$, $j \in \{1, \ldots, k\}$, consider the set of vertices X_j for which v_j is the closest center (including v_j itself), breaking ties arbitrarily. The distance between any two vertices in X_j is at most $3\delta\rho$. Hence the vertices of the bottleneck graph $H_X(3\delta\rho)$ can be partitioned into the sets $X_1, \ldots, X_k$ such that each X_j is a clique in $H_X(3\delta\rho)$. Clearly this also means that each X_j has some graph of $\mathcal{F}$ as a spanning subgraph in $H_X(3\delta\rho)$. Thus we obtain a 3δ-FPA algorithm to the $(k, \mathcal{F})$-PARTITION problem for metrics induced by low highway dimension graphs, with the same asymptotic running time as Algorithm 1.

Note that the reduction would not yield an improved approximation ratio if a 2-approximation was used to solve k-CENTER (which in many cases is the best achievable approximation ratio, as summarized in the introduction), since the radius is always at least half the diameter of a graph, i.e., a $2d$-approximation is already a 4δ-approximation.

6 Open problems

In this last section we summarize some problems left open by our results. The most pressing unanswered question concerns the approximability of k-CENTER using only the highway dimension h as a parameter. Even though we obtained some partial answers in Section 4, these do not exclude the existence of a $(2 - \varepsilon)$-FPA for parameter h alone. Also whether better approximation ratios than $3/2$ can be obtained for the combined parameter (k, h) remains open. In particular, the approximation scheme given by Becker et al. [10] for parameter (k, h) using the more restrictive highway dimension definition as used in [16], makes this an appealing possibility. Even more intriguing would be a hardness result that excludes approximation schemes for parameter (k, h) using the more general Definition 1 for the highway dimension. This would imply that the difference between these definitions is more than just "cosmetic". We note at this point that all lower bound results of Section 4 are applicable to the highway dimension definition used in [10, 16] (only the constants in the sparsity of the shortest path covers increase).

Another interesting open question concerns the computability of the highway dimension. In particular, obtaining better approximation ratios than $O(\log h)$, as given in [2], would improve the running time of not only the k-CENTER algorithm presented here, but also the algorithms given in [10, 16]. This is true even if the running time is parameterized in the highway dimension h. Hence an important question is whether computing the highway dimension is FPT for the canonical

parameter h, or even whether an $o(\log h)$-FPA algorithm exists for this parameter.

Finally, it would also be interesting to see whether the techniques developed here (or in [10, 16]) can be used for other variants of the k-CENTER problem, such as, for instance, the k-SUPPLIER problem [19].

References

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

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

[3] 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):41, 2016.

[4] P. K. Agarwal and C. M. Procopiuc. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2):201–226, 2002.

[5] J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier. Fixed parameter algorithms for dominating set and related problems on planar graphs. *Algorithmica*, 33(4): 461–493, 2002.

[6] P. Alimonti and V. Kann. Some APX-completeness results for cubic graphs. *Theoretical Computer Science*, 237(1):123–134, 2000.

[7] H. Bast, S. Funke, D. Matijevic, P. Sanders, and D. Schultes. In transit to constant time shortest-path queries in road networks. In *ALENEX*, pages 46–59, 2007.

[8] H. Bast, S. Funke, and D. Matijevic. Ultrafast shortest-path queries via transit nodes. *9th DIMACS Implementation Challenge*, 74:175–192, 2009.

[9] R. Bauer, T. Columbus, I. Rutter, and D. Wagner. Search-space size in contraction hierarchies. In *ICALP*, pages 93–104, 2013.

[10] 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. *ArXiv e-prints, arXiv:1707.08270*, 2017.

[11] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

[12] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos. Fixed-parameter algorithms for (k, r)-center in planar graphs and map graphs. *Transactions on Algorithms*, 1(1):33–47, 2005.

[13] R. G. Downey and M. R. Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013.

[14] T. Feder and D. Greene. Optimal algorithms for approximate clustering. In *STOC*, pages 434–444, 1988.

[15] A. E. Feldmann and D. Marx. The parameterized hardness of the k-center problem in transportation networks. In *SWAT*, pages 19:1–19:13, 2018. doi: 10.4230/LIPIcs.SWAT.2018.19.

[16] A. E. Feldmann, W. S. Fung, J. Könemann, and I. Post. A $(1+\epsilon)$-embedding of low highway dimension graphs into bounded treewidth graphs. *SIAM J. Comput.*, 47(4):1667–1704, 2018. doi: 10.1137/16M1067196.

[17] F. Fomin, D. Kratsch, and G. Woeginger. Exact (exponential) algorithms for the dominating set problem. In *WG*, pages 245–256. Springer, 2005.

[18] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *FOCS*, pages 534–543, 2003.

[19] D. S. Hochbaum and D. B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM*, 33(3):533–550, 1986.

[20] W.-L. Hsu and G. L. Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979.

[21] R. Impagliazzo and R. Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

[22] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

[23] I. Katsikarelis, M. Lampis, and V. Paschos. Structural parameters, tight bounds, and approximation for (k,r)-center. In *ISAAC*, pages 50:1–50:13, 2017.

[24] P. Klein. Personal communication, 2017.

[25] A. Kosowski and L. Viennot. Beyond highway dimension: Small distance labels using tree skeletons. In *SODA*, pages 1462–1478, 2017.

[26] D. Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51 (1):60–78, 2008.

[27] D. Marx and M. Pilipczuk. Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams. In *ESA*, pages 865–877. Springer, 2015.

[28] J. Plesník. On the computational complexity of centers locating in a graph. *Aplikace matematiky*, 25(6):445–452, 1980.

[29] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Inc., 2001.

[30] D. P. Williamson and D. B. Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2011. doi: 10.1017/cbo9780511921735.