Structural Parameters, Tight Bounds, and Approximation for (k, r)-Center*

Ioannis Katsikarelis[1], Michael Lampis[2], and Vangelis Th. Paschos[3]

1 Université Paris-Dauphine, PSL Research University, CNRS, UMR 7243, LAMSADE, Paris, France
ioannis.katsikarelis@lamsade.dauphine.fr

2 Université Paris-Dauphine, PSL Research University, CNRS, UMR 7243, LAMSADE, Paris, France
michail.lampis@lamsade.dauphine.fr

3 Université Paris-Dauphine, PSL Research University, CNRS, UMR 7243, LAMSADE, Paris, France
paschos@lamsade.dauphine.fr

Abstract

In (k, r)-Center we are given a (possibly edge-weighted) graph and are asked to select at most k vertices (centers), so that all other vertices are at distance at most r from a center. In this paper we provide a number of tight fine-grained bounds on the complexity of this problem with respect to various standard graph parameters. Specifically:

- For any $r \geq 1$, we show an algorithm that solves the problem in $O^*((3r+1)^{\mathrm{cw}})$ time, where cw is the clique-width of the input graph, as well as a tight SETH lower bound matching this algorithm's performance. As a corollary, for $r = 1$, this closes the gap that previously existed on the complexity of Dominating Set parameterized by cw.
- We strengthen previously known FPT lower bounds, by showing that (k, r)-Center is W[1]-hard parameterized by the input graph's vertex cover (if edge weights are allowed), or feedback vertex set, even if k is an additional parameter. Our reductions imply tight ETH-based lower bounds. Finally, we devise an algorithm parameterized by vertex cover for unweighted graphs.
- We show that the complexity of the problem parameterized by tree-depth is $2^{\Theta(\mathrm{td}^2)}$ by showing an algorithm of this complexity and a tight ETH-based lower bound.

We complement these mostly negative results by providing *FPT approximation schemes* parameterized by clique-width or treewidth which work efficiently independently of the values of k, r. In particular, we give algorithms which, for any $\epsilon > 0$, run in time $O^*((\mathrm{tw}/\epsilon)^{O(\mathrm{tw})})$, $O^*((\mathrm{cw}/\epsilon)^{O(\mathrm{cw})})$ and return a $(k, (1+\epsilon)r)$-center, if a (k, r)-center exists, thus circumventing the problem's W-hardness.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes, F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory

Keywords and phrases FPT algorithms, Approximation, Treewidth, Clique-width, Domination

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2017.50

1 Introduction

In this paper we study the (k, r)-Center problem: given a graph $G = (V, E)$ and a weight function $w : E \rightarrow \mathbb{N}^+$ which satisfies the triangle inequality and defines the length of each

[*] A full version of the paper is available at [25], https://arxiv.org/abs/1704.08868.

28th International Symposium on Algorithms and Computation (ISAAC 2017).
Editors: Yoshio Okamoto and Takeshi Tokuyama; Article No. 50; pp. 50:1–50:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

edge, we are asked if there exists a set K (the *center-set*) of at most k vertices of V, so that $\forall u \in V \setminus K$ we have $\min_{v \in K} d(v, u) \leq r$, where $d(v, u)$ denotes the shortest-path distance from v to u under weight function w. If w assigns weight 1 to all edges we say that we have an instance of un-weighted (k, r)-Center. (k, r)-Center is an extremely well-investigated optimization problem with numerous applications. It has a long history, especially from the point of view of approximation algorithms, where the objective is typically to minimize r for a given k [24, 46, 29, 19, 42, 31, 28, 1, 18]. The converse objective (minimizing k for a given r) has also been well-studied, with the problem being typically called r-Dominating Set in this case [11, 43, 36, 12].

Because (k, r)-Center generalizes Dominating Set (which corresponds to the case $r = 1$), the problem can already be seen to be hard, even to approximate (under standard complexity assumptions). In particular, the optimal r cannot be approximated in polynomial time by a factor better than 2 (even on planar graphs [19]), while k cannot be approximated by a factor better than $\ln n$ [39]. Because of this hardness, we are strongly motivated to investigate the problem's complexity when the input graph has some restricted structure.

In this paper our goal is to perform a complete analysis of the complexity of (k, r)-Center that takes into account this input structure by using the framework of parameterized complexity. In particular, we provide *fine-grained* upper and lower bound results on the complexity of (k, r)-Center with respect to the most widely studied parameters that measure a graph's structure: treewidth **tw**, clique-width **cw**, tree-depth **td**, vertex cover **vc**, and feedback vertex set **fvs**. In addition to the intrinsic value of determining the precise complexity of (k, r)-Center, this approach is further motivated by the fact that FPT algorithms for this problem have often been used as building blocks for more elaborate approximation algorithms [16, 18]. Indeed, (some of) these questions have already been considered, but we provide a number of new results that build on and improve the current state of the art. Along the way, we also close a gap on the complexity of the flagship Dominating Set problem parameterized by clique-width. Specifically, we prove the following:

- (k, r)-Center *can be solved (on unweighted graphs) in time* $O^*((3r + 1)^{cw})$ *(if a clique-width expression is supplied with the input), but it cannot be solved in time* $O^*((3r+1-\epsilon)^{cw})$ *for any (fixed)* $r \geq 1$*, unless the Strong Exponential Time Hypothesis (SETH) [26, 27] fails.* The algorithmic result relies on standard techniques (dynamic programming on clique-width, fast subset convolution), as well as several problem-specific observations which are required to obtain the desired table size. The SETH lower bound follows from a direct reduction from SAT. A noteworthy consequence of our lower bound result is that, *for the case of* Dominating Set*, it closes the gap between the complexity of the best known algorithm (*$O^*(4^{cw})$ *[9]) and the best previously known lower bound (*$O^*((3 - \epsilon)^{cw})$ *[35]).*
- (k, r)-Center *cannot be solved in time* $n^{o(vc+k)}$ *on edge-weighted graphs, or time* $n^{o(fvs+k)}$ *on unweighted graphs, unless the Exponential Time Hypothesis (ETH) is false.* It was already known that an FPT algorithm parameterized just by tw (for unbounded r) is unlikely to be possible [10]. These results show that the same holds for the two more restrictive parameters fvs and vc, even if k is also added as a parameter. They are (asymptotically) tight, since it is easy to obtain $O^*(n^{\text{fvs}})$, $O^*(n^{\text{vc}})$, and $O^*(n^k)$ algorithms. We remark that (k, r)-Center is a rare example of a problem that turns out to be hard parameterized by vc. We complement these lower bounds by an FPT algorithm for the unweighted case, running in time $O^*(5^{\text{vc}})$.
- (k, r)-Center *can be solved in time* $O^*(2^{O(td^2)})$ *for unweighted graphs, but if it can be solved in time* $O^*(2^{o(td^2)})$*, then the ETH is false.* Here the upper bound follows from known connections between a graph's tree-depth and its diameter, while the lower bound

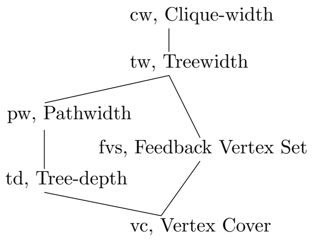

Figure 1 Relationships of parameters. Algorithmic results are inherited downwards, hardness results upwards.

Table 1 A summary of our results (theorem numbers) for all considered parameters. Initials u/w denote the unweighted/weighted variants of the problem.

	cw	tw	fvs	td	vc
FPT exact	3 (w/u)	10 (w/u)		7 (u)	6 (u)
FPT-AS	16 (w/u)	13 (w/u)			
SETH LB	1 (u)				
ETH LB			5 (w/u)	8 (u)	4 (w)
W[1]-hard			5 (w/u)		4 (w)

follows from a reduction from 3-SAT. We remark that this is a somewhat uncommon example of a parameterized problem whose parameter dependence turns out to be exponential in the *square* of the parameter.

These results, together with the recent work of [10] showing tight bounds of $O^*((2r+1)^{\text{tw}})$ on the problem's complexity parameterized by tw, give a complete and often fine-grained, picture on (k,r)-Center for the most important graph parameters. One of the conclusions that can be drawn is that, as a consequence of the problem's hardness for vc (in the weighted case) and fvs, there are few cases where we can hope to obtain an FPT algorithm without bounding r: as r increases the complexity of exactly solving the problem quickly degenerates away from the case of Dominating Set, which is FPT for all considered parameters.

A further contribution of this paper is to complement this negative view by pointing out that it only applies if one insists on solving the problem *exactly.* If we allow algorithms that return a $(1+\epsilon)$-approximation to the optimal r, for arbitrarily small $\epsilon > 0$ and while respecting the given value of k, we obtain the following:

- *There exist algorithms which, for any $\epsilon > 0$, when given a graph that admits a (k,r)-center, return a $(k,(1+\epsilon)r)$-center in time $O^*((tw/\epsilon)^{O(tw)})$, or $O^*((cw/\epsilon)^{O(cw)})$, assuming a tree decomposition or clique-width expression is given in the input.*

The tw approximation algorithm is based on a technique introduced in [32], while the cw algorithm relies on a new extension of an idea from [23], which may be of independent interest. Thanks to these approximation algorithms, we arrive at an improved understanding of the complexity of (k,r)-Center by including the question of approximation, and obtain algorithms which continue to work efficiently even for large values of r. Figure 1 illustrates the relationships between parameters and Table 1 summarizes our results. We refer the reader to the full version [25] for all omitted definitions, constructions and proofs.

Related Work: Our work follows upon recent work by [10], which showed that (k,r)-Center can be solved in $O^*((2r+1)^{\text{tw}})$, but not faster (under SETH), while its connected variant can be solved in $O^*((2r+2)^{\text{tw}})$, but not faster. This paper in turn generalized previous results on Dominating Set for which a series of papers had culminated into an $O^*(3^{\mathrm{tw}})$ algorithm [44, 2, 45], while on the other hand, [35] showed that an $O^*((3-\epsilon)^{\mathrm{pw}})$ algorithm would violate the SETH, where pw denotes the input graph's pathwidth. The complexity of (k, r)-Center by the related parameter branchwidth had previously been considered in [16] where an $O^*((2r+1)^{\frac{3}{2}\mathrm{bw}})$ algorithm is given. Moreover, [37] showed the problem parameterized by the number k of centers to be W[1]-hard in the L_∞ metric, in fact analysing Covering Points with Squares, a geometric variant. It remains W[2]-hard for 2-degenerate graphs [22]. On clique-width, a $O^*(4^{\mathrm{cw}})$-time algorithm for Dominating Set was given in [9], while [41] notes that the lower bound of [35] for pathwidth/treewidth would also imply no $(3-\epsilon)^{\mathrm{cw}} \cdot n^{O(1)}$-time algorithm exists for clique-width under SETH as well, since clique-width is at most 1 larger than pathwidth. For the edge-weighted variant, [20] shows that a $(2-\epsilon)$-approximation is W[2]-hard for parameter k and NP-hard for graphs of *highway dimension* $h = O(\log^2 n)$, while also offering a 3/2-approximation algorithm of running time $2^{O(kh\log(h))} \cdot n^{O(1)}$, exploiting the similarity of this problem with that of solving Dominating Set on graphs of bounded vc. Finally, for unweighted graphs, [34] provides efficient (linear/polynomial) algorithms computing $(r + O(\mu))$-dominating sets and $+O(\mu)$-approximations for (k, r)-Center, where μ is the *tree-breadth* or *cluster diameter* in a layering partition of the input graph, while [18] gives a polynomial-time bicriteria approximation scheme for graphs of bounded genus.

2 Definitions and Preliminaries

We use standard graph-theoretic notation. For a graph $G = (V, E)$, $n = |V|$ denotes the number of vertices, $m = |E|$ the number of edges and for a subset $X \subseteq V$, $G[X]$ denotes the graph induced by X. Further, we assume the reader has some familiarity with standard definitions from parameterized complexity theory, such as the classes FPT, W[1] (see [15, 21, 17]). For a parameterized problem with parameter k, an FPT-AS is an algorithm which for any $\epsilon > 0$ runs in time $O^*(f(k, \frac{1}{\epsilon}))$ (i.e. FPT time when parameterized by $k + \frac{1}{\epsilon}$) and produces a solution at most a multiplicative factor $(1+\epsilon)$ from the optimal (see [38]). We use $O^*(\cdot)$ to imply omission of factors polynomial in n.

In this paper we present approximation schemes with running times of the form $(\log n/\epsilon)^{O(k)}$. These can be seen to imply an FPT running time by a well-known win-win argument (see Lemma 1 in [25]): *If a parameterized problem with parameter k admits, for some $\epsilon > 0$, an algorithm running in time $O^*((\log n/\epsilon)^{O(k)})$, then it also admits an algorithm running in time $O^*((k/\epsilon)^{O(k)})$.*

Treewidth and *pathwidth* are standard notions in parameterized complexity which measure how close a graph is to being a tree or path (see [8, 5, 30]). We will also use the standard graph parameter of *clique-width*, which was introduced as a generalization of treewidth to dense graphs (see [13, 14]). Additionally, we will use the parameters *vertex cover number* and *feedback vertex set number* of a graph G, which are the sizes of the minimum vertex set whose deletion leaves the graph edgeless, or acyclic, respectively. Finally, we will consider the related notion of *tree-depth* [40], which is defined as the minimum height of a rooted forest whose completion (the graph obtained by connecting each node to all its ancestors) contains the input graph as a subgraph. We will denote these parameters for a graph G as $\mathrm{tw}(G), \mathrm{pw}(G), \mathrm{cw}(G), \mathrm{vc}(G), \mathrm{fvs}(G)$, and $\mathrm{td}(G)$, and will omit G if it is clear from the context. We recall the following well-known relations between these parameters [6, 14] which justify the hierarchy given in Figure 1: *For any graph G we have $tw(G) \leq pw(G) \leq td(G) \leq vc(G)$, $tw(G) \leq fvs(G) \leq vc(G)$, $cw(G) \leq pw(G) + 1$, and $cw(G) \leq 2^{tw(G)+1} + 1$.*

We also recall here the two main complexity assumptions used in this paper [26, 27]. The Exponential Time Hypothesis (ETH) states that 3-SAT cannot be solved in time $2^{o(n+m)}$ on instances with n variables and m clauses. The Strong Exponential Time Hypothesis (SETH) states that for all $\epsilon > 0$, there exists an integer k such that k-SAT cannot be solved in time $(2-\epsilon)^n$ on instances of k-SAT with n variables.

3 Clique-width

3.1 Lower bound based on SETH

The result of this section is that for any fixed constant $r \geq 1$, the existence of any algorithm for (k,r)-Center of running time $O^*((3r+1-\epsilon)^{\text{cw}})$, for some $\epsilon > 0$, would imply the existence of some algorithm for SAT of running time $O^*((2-\delta)^n)$, for some $\delta > 0$.

Before we proceed, let us recall the high-level idea behind the SETH lower bound for Dominating Set given in [35], as well its generalization to (k,r)-Center given in [10]. In both cases the key to the reduction is the construction of long paths, which are conceptually divided into blocks of $2r+1$ vertices. The intended solution consists of selecting, say, the i-th vertex of a block of a path, and repeating this selection in all blocks of this path. This allows us to encode $(2r+1)^t$ choices, where t is the number of paths we make, which ends up being roughly equal to the treewidth of the construction. The reason this construction works in the converse direction is that, even though the optimal (k,r)-Center solution may "cheat" by selecting the i-th vertex of a block, and then the j-th vertex of the next, one can see that we must have $j \leq i$. Hence, by making the paths that carry the solution's encoding long enough we can ensure that the solution eventually settles into a pattern that encodes an assignment to the original formula (which can be "read" with appropriate gadgets).

In our lower bound construction for clique-width we need to be able to "pack" more information per unit of width: instead of encoding $(2r+1)$ choices for each unit of treewidth, we need to encode $(3r+1)$ choices for each label. Our high-level plan to achieve this is to use *a pair* of long paths for each label. Because we only want to invest one label for each pair of paths we are forced to periodically (every $2r+1$ vertices) add cross edges between them, so that the connection between blocks can be performed with a single join operation (see the paths A_1, B_1 in Figure 2 for an illustation). Our plan now is to encode a solution by selecting a pair of vertices that will be repeated in each block, for example every i-th vertex of A_1 and every j-th vertex of B_1. One may naively expect that this would allow us to encode $(2r+1)^2$ choices for each label (which would lead to a SETH lower bound that would contradict the algorithm of Section 3.2). However, because of the cross edges, the optimal (k,r)-Center solution is not as well-behaved on a pair of cross-connected paths as it was on a path, and this makes it much harder to execute the converse direction of the reduction: a solution that takes every i-th vertex of A_1 could alternate repeatedly between various choices for B_1, because the selected vertices of A_1 also cover parts of B_1. Our strategy is therefore to identify $(3r+1)$ ordered selection pairs and show that any valid solution must be well-behaved with respect to these pairs. An overview of our construction, omitting most technical details of the reduction's inner mechanism follows.

Construction overview: We construct a graph G, given some $\epsilon < 1$ and an instance ϕ of SAT with n variables and m clauses. We first choose an integer p, depending on ϵ and r (for technical reasons that become apparent in the proof of Theorem 1). Note that for the results of this section, both r and p are considered constants. We then group the variables of ϕ into $t = \lceil \frac{n}{\gamma} \rceil$ groups $F_1, \ldots, F_t$, for $\gamma = \lfloor \log_2 (3r+1)^p \rfloor$, being also the maximum size of any such group. Our graph G will consist of t rows of $m(3rpt+1)$ gadgets $\hat{G}$, each row corresponding to one such group of variables. Each gadget $\hat{G}$ will contain p pairs of paths A_i, B_i and any selection of one vertex from each path will be associated with a specific partial assignment to the variables of the group. Gadgets of the same row will be connected in a path-like manner: for each $i \in [1, p]$ both final vertices of each pair A_i, B_i within each gadget will be connected to both first vertices of the corresponding pair A_i, B_i of the following gadget, with a global vertex h adjacent to all the first/last vertices of all such long paths, with an additional path of length r attached to h to ensure its selection in any minimum-sized center-set (and allowing for any selection in these first/last gadgets to be valid).

Furthermore, we will show that the possible selections of only one vertex from each path can be divided into $3r+1$ equivalence classes: we define $3r+1$ *canonical pairs* of numbers (α_y, β_y), indexed (and ordered) by $y \in [1, 3r+1]$, that give the indices of vertices from a pair of paths A_i, B_i (i.e. the α_y-th vertex of A_i and the β_y-th vertex of B_i) that would form the characteristic selection for each class, and show that any other selection within each class would be interchangeable (in terms of domination/coverage) with the characteristic selection, while if some pair with index y is used for selection of vertices from paths A_i, B_i in some gadget $\hat{G}_\tau^j$, then any pair used for the paths of the following gadget $\hat{G}_\tau^{j+1}$ (on the same row) must be of index $y' \leq y$. Observe that, as the path selections from each column must be well behaved with respect to our canonical pairs, there is an upper bound of $3r$ on the number of times the selection pattern can change on some pair of paths, giving $3rp$ for each row of gadgets and $3rpt$ times overall. In each gadget $\hat{G}$, we also make $3r+1$ vertices u_i^y for each pair A_i, B_i that signify these canonical selections from each path and further, a group of $(3r+1)^p$ vertices x_S for each set S that only contains one such u_i^y for each $i \in [1, p]$.

In this way, a selection of one vertex from each path A_i, B_i will correspond to a selection u_i^y, while all p such selections will correspond to one selection x_S that will in turn be associated with a partial assignment to the group of variables assigned to this row of gadgets (there are 2^γ partial assignments for each group and $(3r+1)^p \geq 2^\gamma$ sets S). Further, each column of gadgets will correspond to one clause, with the first m columns assigned to one clause each and $3rpt+1$ repetitions of this pattern giving the complete association. Our graph G will have one vertex $\hat{c}$ for each such column of gadgets (representing the associated clause) at distance r from vertices x_S in the gadgets $\hat{G}$ of its column that represent the partial assignments to the variables of the group associated with the gadget's row (and group F_τ) that would satisfy the clause (Figure 2 provides an illustration).

Thus, a satisfying assignment for ϕ will give a (k, r)-center for G by selecting in each gadget $\hat{G}$ all vertices corresponding to the partial assignment for its associated group of variables from each pair of paths, as well as the matching u_i^y and x_S vertices (and h). For the converse direction, as the number of changes of selection pattern is $\leq 3rpt$ and the number of columns is $m(3rpt+1)$, by the pigeonhole principle, there will always exist m consecutive columns for which the pattern does not change and thus we will be able to extract a consistent assignment for all clauses.

▶ **Theorem 1.** *For any fixed $r \geq 1$, if (k, r)-*Center *can be solved in $O^*((3r+1-\epsilon)^{cw(G)})$ time for some $\epsilon > 0$, then* SAT *can be solved in $O^*((2-\delta)^n)$ time for some $\delta > 0$.*

▶ **Corollary 2.** *If* Dominating Set *can be solved in $O^*((4-\epsilon)^{cw(G)})$ time for some $\epsilon > 0$, then* SAT *can be solved in $O^*((2-\delta)^n)$ time for some $\delta > 0$.*

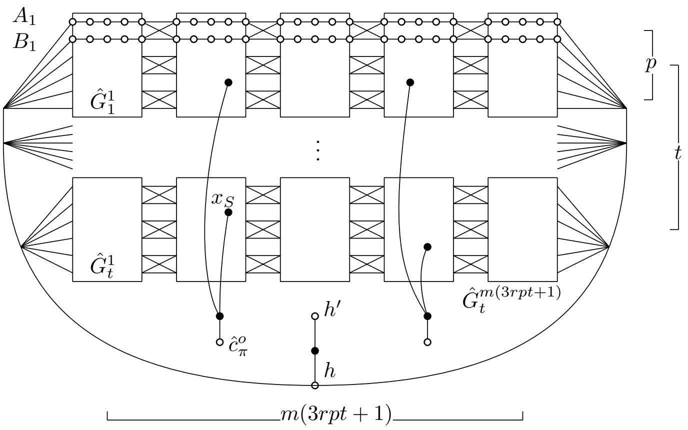

Figure 2 A simplified picture of the complete construction. Note boxes indicate block gadgets $\hat{G}$, while there is no vertex anywhere between h and the first/last vertices of the long paths.

3.2 Dynamic programming algorithm

We next present an $O^*((3r+1)^{\mathrm{cw}})$-time dynamic programming (DP) algorithm for unweighted (k,r)-Center, using a given clique-width expression T_G for G with at most cw labels. Even though the algorithm relies on standard techniques, there are several non-trivial, problem-specific observations that we need to make to reach a DP table size of $(3r+1)^{\mathrm{cw}}$.

Our first step is to re-cast the problem as a *distance-labeling* problem (not be confused with 'label'/'label-set' for a clique-width expression), that is, to formulate the problem as that of deciding for each vertex what is its precise distance to the optimal solution K. This is helpful because it allows us to make the constraints of the problem local, and hence easier to verify: roughly speaking, we say that a vertex is satisfied if it has a neighbor with a smaller distance to K. It is now not hard to design a clique-width based DP algorithm for this version of the problem: for each label l we need to remember two numbers, namely the smallest distance value given to some vertex with label l, and the smallest distance value given to a *currently unsatisfied* vertex with label l, if it exists.

The above scheme directly leads to an algorithm running in time (roughly) $((r+1)^2)^{\mathrm{cw}}$. In order to decrease the size of this table, we now make the following observation: if a label-set contains a vertex at distance i from K, performing a join operation will satisfy all vertices that expect to be at distance $\geq i+2$ from K, since all vertices of the label-set will now be at distance at most 2. This implies that, in a label-set where the minimum assigned value is i, states where the minimum unsatisfied value is between $i+2$ and r are effectively equivalent. With this observation we can bring down the size of the table to $(4r)^{\mathrm{cw}}$, because (intuitively) there are four cases for the smallest unsatisfied value: $i, i+1, \geq i+2$, and the case where all values are satisfied.

The last trick that we need to achieve the promised running time departs slightly from the standard DP approach. We will say that a label-set is *live* in a node of the clique-width expression if there are still edges to be added to the graph that will be incident to its vertices. During the execution of the dynamic program, we perform a "fore-tracking" step, by checking the part of the graph that comes higher in the expression to determine if a label-set is live. If it is, we merge the case where the smallest unsatisfied value is $i+2$, with the case where all values are satisfied (since a join operation will eventually be performed). Otherwise, a partial solution that contains unsatisfied vertices in a non-live label-set can safely be discarded. This brings down the size of the DP table to $(3r+1)^{\mathrm{cw}}$, and then we need to use some further techniques to make the total running time quasi-linear in the size of the table. This involves counting the number of solutions instead of directly computing a solution of minimum size, as well as a non-trivial extension of fast subset convolution from [4] for a $3 \times (r + 1)$-sized table (or *state-changes*, see [45, 9] and Chapter 11 of [15]).

▶ **Theorem 3.** *Given graph G, along with $k, r \in \mathbb{N}^+$ and clique-width expression T_G of clique-width cw for G, there exists an algorithm to solve the counting version of the (k, r)-*Center *problem in $O^*((3r + 1)^{cw})$ time.*

4 Vertex Cover, Feedback Vertex Set and Tree-depth

In this section we first show that the edge-weighted variant of the (k, r)-Center problem parameterized by $\mathrm{vc} + k$ is W[1]-hard, and more precisely, that the problem does not admit a $n^{o(\mathrm{vc}+k)}$ algorithm under the ETH. We give a reduction from k-Multicolored Independent Set.

This is a well-known W[1]-hard problem that cannot be solved in $n^{o(k)}$ under the ETH [15]. Using essentially the same reduction with that of Theorem 4, we obtain a similar hardness result for unweighted (k, r)-Center parameterized by fvs.

▶ **Theorem 4.** *The weighted (k, r)-*Center *problem is W[1]-hard parameterized by $vc + k$. Furthermore, if there is an algorithm for weighted (k, r)-*Center *running in time $n^{o(vc+k)}$ then the ETH is false.*

▶ **Corollary 5.** *The (k, r)-*Center *problem is W[1]-hard when parameterized by $fvs + k$. Furthermore, if there is an algorithm for weighted (k, r)-*Center *running in time $n^{o(fvs+k)}$, then the ETH is false.*

We next show that unweighted (k, r)-Center admits an algorithm running in time $O^*(5^{\mathrm{vc}})$, in contrast to its weighted version (Theorem 4). We devise an algorithm that operates in two stages: first, it guesses the intersection of the optimal solution with the optimal vertex cover, and then it uses a reduction to Set Cover to complete the solution optimally.

▶ **Theorem 6.** *Given graph G, along with $k, r \in \mathbb{N}^+$ and a vertex cover of size vc of G, there exists an algorithm solving unweighted (k, r)-*Center *in $O^*(5^{vc})$ time.*

We next consider the un-weighted version of (k, r)-Center parameterized by td. Theorem 4 has estabilshed that weighted (k, r)-Center is W[1]-hard parameterized by vc (and hence also by td), but the complexity of unweighted (k, r)-Center parameterized by td does not follow from this theorem, since td is incomparable to fvs. Indeed, we show that (k, r)-Center is FPT parameterized by td and precisely determine its parameter dependence based on the ETH.

▶ **Theorem 7.** *Unweighted (k, r)-*Center *can be solved in time $O^*(2^{O(td^2)})$.*

▶ **Theorem 8.** *If (k, r)-*Center *can be solved in $2^{o(td^2)} \cdot n^{O(1)}$ time, then* 3-SAT *can be solved in $2^{o(n)}$ time.*

5 Treewidth: FPT approximation scheme

In this section we present an FPT approximation *scheme* (FPT-AS) for (k, r)-Center parameterized by tw. Given as input a weighted graph $G = (V, E)$, $k, r \in \mathbb{N}^+$ and an arbitrarily small error parameter $\epsilon > 0$, our algorithm is able to return a solution that uses a set of k centers K, such that all other vertices are at distance at most $(1+\epsilon)r$ from K, or to correctly conclude that no (k,r)-center exists. The running time of the algorithm is $O^*((\mathrm{tw}/\epsilon)^{O(\mathrm{tw})})$, which (for large r) significantly out-performs any *exact* algorithm for the problem (even for the unweighted case and more restricted parameters, as in Theorems 4 and 5), while only sacrificing a small ϵ error in the quality of the solution.

Our algorithm will rely heavily on a technique introduced in [32] (see also [3]) to approximate problems which are W-hard by treewidth. The idea is that, if the hardness of the problem is due to the fact that the DP table needs to store tw large numbers (in our case, the distances of the vertices in the bag from the closest center), we can significantly speed up the algorithm if we replace all entries by the closest integer power of $(1+\delta)$, for some appropriately chosen δ. This will reduce the table size from (roughly) r^{tw} to $(\log_{(1+\delta)} r)^{\mathrm{tw}}$.

The problem now is that a DP performing calculations on its entries will, in the course of its execution, create values which are not integer powers of $(1+\delta)$, and will therefore have to be “rounded” to retain the table size. This runs the risk of accumulating rounding errors, but we manage to show that the error on any entry of the rounded table can be bounded by a function of the height of its corresponding bag, then using a theorem of [7] stating that any tree decomposition can be balanced so that its width remains almost unchanged, yet its total height becomes $O(\log n)$. Beyond these ideas, which are for the most part present in [32], we will also need a number of problem-specific observations, such as the fact that we can pre-process the input by taking the metric closure of each bag, and in this way avoid some error-prone arithmetic operations.

To obtain the promised algorithm we thus do the following: first we re-cast the problem as a distance-labeling problem (as in the proof of Theorem 3) and formulate an exact treewidth-based DP algorithm running in time $O^*(r^{O(\mathrm{tw})})$. We remark that the algorithm essentially reproduces the ideas of [10], and can be made to run in $O^*((2r+1)^{\mathrm{tw}})$ if one uses fast subset convolution for the Join nodes (the naive implementation would need time $O^*((2r+1)^{2\mathrm{tw}}))$ but we give it here to ensure that we have a solid foundation upon which to build the approximation algorithm. We then apply the rounding procedure sketched above, and prove its approximation ratio by using the balancing theorem of [7] and indirectly comparing the value produced by the approximation algorithm with the value that would have been produced by the exact algorithm.

Distance-labeling: We give an equivalent formulation of (k,r)-Center that will be more convenient to work with in the remainder, similarly to Section 3.2. For an edge-weighted graph $G=(V,E)$, a distance-labeling function is a function $\mathrm{dl}: V \rightarrow \{0,\dots,r\}$. We say that $u \in V$ is *satisfied* by dl, if $\mathrm{dl}(u)=0$, or if there exists $v \in N(u)$ such that $\mathrm{dl}(u) \geq dl(v) + w((v,u))$. We say that dl is *valid* if all vertices of V are satisfied by dl, and we define the *cost* of dl as $|\mathrm{dl}^{-1}(0)|$. The following lemma shows the equivalence between the two formulations:

▶ **Lemma 9.** *An edge-weighted graph $G=(V,E)$ admits a (k,r)-center if and only if it admits a valid distance-labeling function $dl: V \rightarrow \{0,\dots,r\}$ with cost k.*

▶ **Theorem 10.** *There is an algorithm which, given an edge-weighted graph $G=(V,E)$ and $r \in \mathbb{N}^+$, computes the minimum cost of any valid distance labeling of G in time $O^*(r^{O(tw)})$.*

We now describe an approximation algorithm based on the exact DP algorithm of Theorem 10. We make use of a result of [7] stating that: *There is an algorithm which, given a tree decomposition of width w of a graph on n nodes, produces a decomposition of the same graph with width at most $3w+2$ and height $O(\log n)$ in polynomial time* and of the following lemma:

▶ **Lemma 11.** *Let $G = (V, E)$ be an edge-weighted graph, $\mathcal{T}$ a tree decomposition of G, and $u, v \in V$ two vertices that appear together in a bag of $\mathcal{T}$. Let G' be the graph obtained from G by adding (u, v) to E (if it does not already exist) and setting $w((u, v)) = d_G(u, v)$. Then $\mathcal{T}$ is a valid decomposition of G', and $\forall k, r$, G' admits a (k, r)-center if and only if G does.*

Let us also give an approximate version of the distance labeling problem we defined above, for a given error parameter $\epsilon > 0$. Let $\delta > 0$ be some appropriately chosen secondary parameter (we will eventually set $\delta \approx \frac{\epsilon}{\log n}$). We define a δ-*labeling* function dl_δ as a function from V to $\{0\} \cup \{(1+\delta)^i \mid i \in \mathbb{N}, (1+\delta)^i \leq (1+\epsilon)r\}$. In words, such a function assigns (as previously) a distance label to each vertex, with the difference that now all values assigned are integer powers of $(1+\delta)$, and the maximum value is at most $(1+\epsilon)r$. We now say that a vertex u is ϵ-*satisfied* if $\mathrm{dl}_\delta(u) = 0$ or, for some $v \in N(u)$ we have $\mathrm{dl}_\delta(u) \geq \mathrm{dl}_\delta(v) + \frac{w((v,u))}{1+\epsilon}$. As previously, we say that dl_δ is *valid* if all vertices are ϵ-satisfied, and define its cost as $|\mathrm{dl}_\delta^{-1}(0)|$. The following Lemma 12 shows the equivalence of a valid δ-labeling function of cost k and a $(k, (1+\epsilon)^2 r)$-center for G and using it we conclude the proof of Theorem 13, stating the main result of this section.

▶ **Lemma 12.** *If for a weighted graph $G = (V, E)$ and any $k, r, \delta, \epsilon > 0$, there exists a valid δ-labeling function with cost k, then there exists a $(k, (1+\epsilon)^2 r)$-center for G.*

▶ **Theorem 13.** *There is an algorithm which, given a weighted instance of (k, r)-*Center*, $[G, k, r]$, a tree decomposition of G of width tw and a parameter $\epsilon > 0$, runs in time $O^*((tw/\epsilon)^{O(tw)})$ and either returns a $(k, (1+\epsilon))$-center of G, or correctly concludes that G has no (k, r)-center.*

6 Clique-width revisited: FPT approximation scheme

We give here an FPT-AS for (k, r)-Center parameterized by cw, both for un-weighted and for weighted instances (for a weighted definition of cw which we explain below). Our algorithm builds on the algorithm of Section 5, and despite the added generality of the parameterization by cw, we are able to obtain an algorithm with similar performance: for any $\epsilon > 0$, our algorithm runs in time $O^*((\mathrm{cw}/\epsilon)^{O(\mathrm{cw})})$ and produces a $(k, (1+\epsilon)r)$-center if the input instance admits a (k, r)-center.

Our main strategy, which may be of independent interest, is to pre-process the input graph $G = (V, E)$ in such a way that the answer does not change, yet producing a graph whose tw is bounded by $O(\mathrm{cw}(G))$. The main insight that we rely on, which was first observed by [23], is that a graph of low cw can be transformed into a graph of low tw if one removes all large bi-cliques. Unlike previous applications of this idea (e.g. [33]), we do not use the main theorem of [23] as a "black box", but rather give an explicit construction of a tree decomposition, exploiting the fact that (k, r)-Center allows us to relatively easily eliminate complete bi-cliques. As a result, we obtain a tree decomposition of width not just bounded by some function of $\mathrm{cw}(G)$, but actually $O(\mathrm{cw}(G))$.

In the remainder we deal with the weighted version of (k, r)-Center. To allow clique-width expressions to handle weighted edges, we interpret the clique-width join operation η as taking three arguments. The interpretation is that $\eta(a, b, w)$ is an operation that adds (directed) edges from all vertices with label a to all vertices with label b and gives weight w to all these edges. It is not hard to see that if a graph has a (standard) clique-width expression with cw labels, it can also be constructed with cw labels in our context, if we replace every standard join operation $\eta(a, b)$ with $\eta(a, b, 1)$ followed by $\eta(b, a, 1)$. Hence, the algorithm we give also applies to un-weighted instances parameterized by (standard) clique-width. We will also deal with a generalization of (k, r)-Center, where we are also supplied, along with the input graph $G = (V, E)$, a subset $I \subseteq V$ of *irrelevant* vertices. In this version, a (k, r)-center is a set $K \subseteq V \setminus I$, with $|K| = k$, such that all vertices of $V \setminus I$ are at distance at most r from K. Clearly, the standard version of (k, r)-Center corresponds to $I = \emptyset$. As we explain in the proof of Theorem 16, this generalization does not make the problem significantly harder. In addition to the above, in this section we allow edge weights to be equal to 0. This does not significantly alter the problem, however, if we are interested in approximation and allow r to be unbounded, as the following lemma shows:

▶ **Lemma 14.** *There exists a polynomial algorithm which, for any $\epsilon > 0$, given an instance $I = [G, w, k, r]$ of (k, r)-Center, with weight function $w : V \to \mathbb{N}$, produces an instance $I' = [G, w', k, r']$ on the same graph with weight function $w' : V \to \mathbb{N}^+$, such that we have the following: for any $\rho \geq 1$, any $(k, \rho r')$-center of I' is a $(k, \rho r)$-center of I; any $(k, \rho r)$-center of I is a $(k, (1 + \epsilon)\rho r')$-center of I'.*

Our main tool is the following lemma, whose strategy is to replace every large label-set by two "representative" vertices, in a way that retains the same distances among all vertices of the graph. Applying this transformation repeatedly results in a graph with small treewidth. The main theorem of this section then follows from the above.

▶ **Lemma 15.** *Given a (k, r)-Center instance $G = (V, E)$ along with a clique-width expression T for G on cw labels, we can in polynomial time obtain a (k, r)-Center instance $G' = (V', E')$ with $V \subseteq V'$, and a tree decomposition of G' of width $tw = O(cw)$, with the following property: for all k, r, G has a (k, r)-center if and only if G' has a (k, r)-center.*

▶ **Theorem 16.** *Given $G = (V, E)$, $k, r \in \mathbb{N}^+$, clique-width expression T for G on cw labels and $\epsilon > 0$, there exists an algorithm that runs in time $O^*((cw/\epsilon)^{O(cw)})$ and either produces a $(k, (1 + \epsilon)r)$-center, or correctly concludes that no (k, r)-center exists.*

References

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

2 J. Alber and R. Niedermeier. Improved tree decomposition based algorithms for domination-like problems. In *LATIN*, volume 2286 of *LNCS*, pages 613–628, 2002.

3 E. Angel, E. Bampis, B. Escoffier, and M. Lampis. Parameterized power vertex cover. In *WG*, volume 9941 of *LNCS*, pages 97–108, 2016.

4 A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets möbius: fast subset convolution. In *STOC*, pages 67–74, 2007.

5 H.L. Bodlaender. Treewidth: Characterizations, applications, and computations. In *WG*, volume 4271 of *LNCS*, pages 1–14. Springer, 2006.

6 H.L. Bodlaender, J.R. Gilbert, H. Hafsteinsson, and T. Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *J. Algorithms*, 18(2):238–255, 1995.

7 H.L. Bodlaender and T. Hagerup. Parallel algorithms with optimal speedup for bounded treewidth. *SIAM J. Comput.*, 27(6):1725–1746, 1998.

8 H.L. Bodlaender and A.M.C.A. Koster. Combinatorial optimization on graphs of bounded treewidth. *Comput. J.*, 51(3):255–269, 2008.

9 H.L. Bodlaender, E.J. van Leeuwen, J.M.M. van Rooij, and M. Vatshelle. Faster algorithms on branch and clique decompositions. In *MFCS*, volume 6281 of *LNCS*, pages 174–185, 2010.

10 G. Borradaile and H. Le. Optimal dynamic program for r-domination problems over tree decompositions. In *IPEC*, volume 63, pages 8:1–8:23, 2016.

11 A. Brandstädt and F.F. Dragan. A linear-time algorithm for connected r-domination and steiner tree on distance-hereditary graphs. *Networks*, 31(3):177–182, 1998.

12 R.S. Coelho, P.F. S. Moura, and Y. Wakabayashi. The k-hop connected dominating set problem: hardness and polyhedra. *Electronic Notes in Discrete Mathematics*, 50:59–64, 2015.

13 B. Courcelle, J.A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

14 B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.

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

16 E.D. Demaine, F.V. Fomin, M.T. Hajiaghayi, and D.M. Thilikos. Fixed-parameter algorithms for (k, r)-center in planar graphs and map graphs. *ACM Trans. Algorithms*, 1(1):33–47, 2005.

17 R.G. Downey and M.R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013.

18 D. Eisenstat, P.N. Klein, and C. Mathieu. Approximating k-center in planar graphs. In *SODA*, pages 617–627, 2014.

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

20 A.E. Feldmann. Fixed Parameter Approximations for k-Center Problems in Low Highway Dimension Graphs. In *ICALP*, volume 9135 of *LNCS*, 2015.

21 J. Flum and M. Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.

22 P.A. Golovach and Y. Villanger. Parameterized Complexity for Domination Problems on Degenerate Graphs. In *WG*, volume 5344 of *LNCS*, page 195–205, 2008.

23 F. Gurski and E. Wanke. The tree-width of clique-width bounded graphs without $K_{n,\ n}$. In *WG*, volume 1928 of *LNCS*, pages 196–205, 2000.

24 D.S. Hochbaum and D.B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *J. ACM*, 33(3):533–550, 1986.

25 I.Katsikarelis, M. Lampis, and V.Th. Paschos. Structural parameters, tight bounds, and approximation for (k, r)-center. *CoRR*, abs/1704.08868, 2017.

26 R. Impagliazzo and R. Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

27 R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

28 S. Khuller, R. Pless, and Y.J. Sussmann. Fault tolerant k-center problems. *Theor. Comput. Sci.*, 242(1-2):237–245, 2000.

29 S. Khuller and Y.J. Sussmann. The capacitated K-center problem. *SIAM J. Discrete Math.*, 13(3):403–418, 2000.

30 T. Kloks. *Treewidth, Computations and Approximations*, volume 842 of *LNCS*. Springer, 1994.

31 S.O. Krumke. On a generalization of the p-center problem. *Inf. Process. Lett.*, 56(2):67–71, 1995.

32 M. Lampis. Parameterized approximation schemes using graph widths. In *ICALP*, volume 8572 of *LNCS*, pages 775–786. Springer, 2014.

33 M. Lampis, K. Makino, V. Mitsou, and Y. Uno. Parameterized edge hamiltonicity. In *WG*, volume 8747 of *LNCS*, pages 348–359, 2014.

34 A. Leitert and F.F. Dragan. Parameterized approximation algorithms for some location problems in graphs. *CoRR*, abs/1706.07475v1, 2017.

35 D. Lokshtanov, D. Marx, and S. Saurabh. Known algorithms on graphs on bounded treewidth are probably optimal. In *SODA*, pages 777–789, 2011.

36 D. Lokshtanov, N. Misra, G. Philip, M.S. Ramanujan, and S. Saurabh. Hardness of r-dominating set on graphs of diameter (r + 1). In *IPEC*, volume 8246 of *LNCS*, pages 255–267, 2013.

37 D. Marx. Efficient approximation schemes for geometric problems? In *ESA*, volume 3669 of *LNCS*, pages 448–459, 2005.

38 D. Marx. Parameterized complexity and approximation algorithms. *Comput. J.*, 51(1):60–78, 2008.

39 D. Moshkovitz. The projection games conjecture and the NP-hardness of ln n-approximating set-cover. *Theory of Computing*, 11:221–235, 2015.

40 J. Nesetril and P. Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *Eur. J. Comb.*, 27(6):1022–1041, 2006.

41 S. Oum, S.H. Sæther, and M. Vatshelle. Faster algorithms for vertex partitioning problems parameterized by clique-width. *Theor. Comput. Sci.*, 535:16–24, 2014.

42 R. Panigrahy and S. Vishwanathan. An o(log* n) approximation algorithm for the asymmetric *p*-center problem. *J. Algorithms*, 27(2):259–268, 1998.

43 P.J. Slater. R-domination in graphs. *J. ACM*, 23(3):446–450, 1976.

44 J.A. Telle and A. Proskurowski. Practical algorithms on partial k-trees with an application to domination-like problems. In *WADS*, volume 709 of *LNCS*, pages 610–621, 1993.

45 J.M.M. van Rooij, H.L. Bodlaender, and P. Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In *ESA*, volume 5757 of *LNCS*, pages 566–577, 2009.

46 V.V. Vazirani. *Approximation algorithms.* Springer, 2001.