Structural Parameterizations for Two Bounded Degree Problems Revisited

Michael Lampis
Université Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016, Paris, France

Manolis Vasilakis
Université Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016, Paris, France

Abstract

We revisit two well-studied problems, Bounded Degree Vertex Deletion and Defective Coloring, where the input is a graph G and a target degree Δ and we are asked either to edit or partition the graph so that the maximum degree becomes bounded by Δ. Both problems are known to be parameterized intractable for the most well-known structural parameters, such as treewidth.

We revisit the parameterization by treewidth, as well as several related parameters and present a more fine-grained picture of the complexity of both problems. In particular:

- Both problems admit straightforward DP algorithms with table sizes $(\Delta+2)^{\mathrm{tw}}$ and $(\chi_{\mathrm{d}}(\Delta+1))^{\mathrm{tw}}$ respectively, where tw is the input graph's treewidth and χ_{d} the number of available colors. We show that, under the SETH, both algorithms are essentially optimal, for any non-trivial fixed values of $\Delta, \chi_{\mathrm{d}}$, even if we replace treewidth by pathwidth. Along the way, we obtain an algorithm for Defective Coloring with complexity quasi-linear in the table size, thus settling the complexity of both problems for treewidth and pathwidth.
- Given that the standard DP algorithm is optimal for treewidth and pathwidth, we then go on to consider the more restricted parameter tree-depth. Here, previously known lower bounds imply that, under the ETH, Bounded Vertex Degree Deletion and Defective Coloring cannot be solved in time $n^{o(\sqrt[4]{\mathrm{td}})}$ and $n^{o(\sqrt{\mathrm{td}})}$ respectively, leaving some hope that a qualitatively faster algorithm than the one for treewidth may be possible. We close this gap by showing that neither problem can be solved in time $n^{o(\mathrm{td})}$, under the ETH, by employing a recursive low tree-depth construction that may be of independent interest.
- Finally, we consider a structural parameter that is known to be restrictive enough to render both problems FPT: vertex cover. For both problems the best known algorithm in this setting has a super-exponential dependence of the form $\mathrm{vc}^{\mathcal{O}(\mathrm{vc})}$. We show that this is optimal, as an algorithm with dependence of the form $\mathrm{vc}^{o(\mathrm{vc})}$ would violate the ETH. Our proof relies on a new application of the technique of d-detecting families introduced by Bonamy et al. [ToCT 2019].

Our results, although mostly negative in nature, paint a clear picture regarding the complexity of both problems in the landscape of parameterized complexity, since in all cases we provide essentially matching upper and lower bounds.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases ETH, Parameterized Complexity, SETH

Digital Object Identifier 10.4230/LIPIcs.ESA.2023.49

Related Version *Full Version*: https://arxiv.org/abs/2304.14724

Funding This work is partially supported by ANR projects ANR-21-CE48-0022 (S-EX-AP-PE-AL) and ANR-18-CE40-0025-01 (ASSK).

31st Annual European Symposium on Algorithms (ESA 2023).
Editors: Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman; Article No. 49; pp. 49:1–49:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Parameterized complexity and in particular the study of structural parameters such as treewidth is one of the most well-developed approaches for dealing with NP-hard problems on graphs. Treewidth is of course one of the major success stories of this field, as a plethora of hard problems become fixed-parameter tractable when parameterized by this parameter. Naturally, this success has motivated the effort to trace the limits of the algorithmic power of treewidth by attempting to understand what are the problems for which treewidth-based techniques *cannot* work.

When could the treewidth toolbox fail? One common scenario that seems to be shared by a multitude of problems which are W[1]-hard[1] parameterized by treewidth is when a natural dynamic programming algorithm does exist, but the DP is forced to store for each vertex of a bag in the tree decomposition an arbitrarily large integer – for example a number related to the degree of the vertex. Our goal in this paper is to study situations of this type and pose the natural question of whether one can do better than the “obvious” DP, by obtaining an algorithm with better running time, even at the expense of looking at a parameter more restrictive than treewidth.

Given the above, we focus on two problems which are arguably among the most natural representatives of our scenario: Bounded Degree Vertex Deletion and Defective Coloring. In both problems the input is a graph G and a target degree Δ and we are asked, in the case of Bounded Degree Vertex Deletion to delete a minimum number of vertices so that the remaining graph has degree at most Δ, and in the case of Defective Coloring to partition G into a minimum number of color classes such that each class induces a graph of degree at most Δ. Both problems are well-studied, as they generalize classical problems (Vertex Cover and Coloring respectively) and we review some of the previous work below. However, the most relevant aspect of the two problems for our purposes is the following: (i) both problems admit DP algorithms with complexity of the form $n^{\mathcal{O}(\mathrm{tw})}$ and (ii) both problems are W[1]-hard parameterized by treewidth; in fact, for Defective Coloring, it is even known that assuming the ETH it cannot be solved in time $n^{o(\mathrm{tw})}$ [5].

Since the $n^{\mathcal{O}(\mathrm{tw})}$ algorithms follow from standard DP techniques, it becomes a natural question whether we can do better. Does a better algorithm exist? Realistically, one could hope for one of two things: either an algorithm which still handles the problem parameterized by treewidth and in view of the aforementioned lower bound only attempts a fine-grained improvement in the running time; or an algorithm which is qualitatively faster at the expense of using a more restricted parameter. The results of this paper give strong negative evidence for both questions: if we parameterize by treewidth (and even by pathwidth) the running time of the standard DP is optimal under the SETH even for all fixed values of the other relevant parameters (Δ and the number of colors χ_{d}); while if we parameterize by more restrictive parameters, such as tree-depth and vertex cover, we obtain lower bound results (under the ETH) which indicate that the best algorithm is still essentially to run a form of the standard treewidth DP, even in these much more restricted cases. Our results thus paint a complete picture of the structurally parameterized complexity of these two problems and indicate that the standard DP is optimal in a multitude of restricted cases.

[1] We assume the reader is familiar with the basics of parameterized complexity theory, as given in standard textbooks [14].

Our contribution in more detail. Following standard techniques, the two problems admit DP algorithms with tables of sizes $(\Delta+2)^{\text{tw}}$ and $(\chi_{\text{d}}(\Delta+1))^{\text{tw}}$ respectively. Our first result is a collection of reductions proving that, assuming the SETH, no algorithm can improve upon these dynamic programs, even for pathwidth. More precisely, we show that no algorithm can solve Bounded Degree Vertex Deletion and Defective Coloring in time $(\Delta+2-\varepsilon)^{\text{pw}}n^{\mathcal{O}(1)}$ and $(\chi_{\text{d}}(\Delta+1)-\varepsilon)^{\text{pw}}n^{\mathcal{O}(1)}$ respectively, for any $\varepsilon>0$ and for any combination of fixed values of Δ, χ_{d} (except the combination $\Delta=0$ and $\chi_{\text{d}}=2$, which trivially makes Defective Coloring polynomial-time solvable). Our reductions follow the general strategy pioneered by Lokshtanov, Marx, and Saurabh [37] and indeed generalize their results for Vertex Cover and Coloring (which already covered the case $\Delta=0$). The main difficulty here is being able to cover all values of the secondary parameters and for technical reasons we are forced to give separate versions of our reductions to cover the case $\Delta=1$ for both problems. Along the way we note that, even though an algorithm with complexity $(\chi_{\text{d}}\Delta)^{\mathcal{O}(\text{tw})}n^{\mathcal{O}(1)}$ was given for Defective Coloring in [5], it was not known if an algorithm with complexity $(\chi_{\text{d}}(\Delta+1))^{\text{tw}}n^{\mathcal{O}(1)}$ (that is, with a quasi-linear dependence on the table size) is possible. For completeness, we settle this by providing an algorithm of this running time, using the FFT technique proposed by Cygan and Pilipczuk [17]. Taking also into account the Bounded Degree Vertex Deletion algorithm of running time $(\Delta+2)^{\text{tw}}n^{\mathcal{O}(1)}$ given by van Rooij [48], we have exactly matching upper and lower bounds for both problems, for both treewidth and pathwidth.

Given that the results above show rather conclusively that the standard DP is the best algorithm for parameters treewidth and pathwidth, we then move on to a more restricted case: tree-depth. We recall that graphs of tree-depth k are a proper subclass of graphs of pathwidth k, therefore one could reasonably hope to obtain a better algorithm for this parameter. This hope may further be supported by the fact that known lower bounds do not match the complexity of the standard algorithm. More precisely, the W[1]-hardness reduction given for Bounded Degree Vertex Deletion parameterized by tree-depth by Ganian, Klute, and Ordyniak [27] has a quartic blow-up, thus only implying that no $n^{o(\sqrt[4]{\text{td}})}$ algorithm is possible; while the reduction given for Defective Coloring in [5] has a quadratic blow-up, only implying that no $n^{o(\sqrt{\text{td}})}$ algorithm is possible (in both cases under the ETH). Our contribution is to show that both reductions can be replaced by more efficient reductions which are *linear* in the parameter; we thus establish that neither problem can be solved in time $n^{o(\text{td})}$, implying that the treewidth-based algorithm remains (qualitatively) optimal even in this restricted case. One interesting aspect of our reductions is that, rather than using a modulator to a low tree-depth graph, which is common in such reductions, we use a recursive construction that leverages the full power of the parameter and may be of further use in tightening other lower bounds for the parameter tree-depth.

Finally, we move on to a more special case, parameterizing both problems by vertex cover. Both problems are FPT for this parameter and, since vertex cover is very restrictive as a parameter, one would hope that, finally, we should be able to obtain an algorithm that is more clever than the treewidth-based DP. Somewhat disappointingly, the known FPT algorithms for both problems have complexity $\text{vc}^{\mathcal{O}(\text{vc})}n^{\mathcal{O}(1)}$ [5], and the super-exponential dependence on the parameter is due to the fact that both algorithms are simple win/win arguments which, in one case, just execute the standard treewidth DP. We show that this is justified, as neither problem can be solved in time $\text{vc}^{o(\text{vc})}n^{\mathcal{O}(1)}$ (under the ETH), meaning that the algorithm that blindly executes the treewidth-based DP in some cases is still (qualitatively) best possible. We obtain our result by applying the technique of d-detecting families, introduced by Bonamy et al. [10]. Our results indicate that parameterization by vertex cover is a domain where this promising, but currently under-used, technique may find more applications in parameterized complexity.

Related work. Both Bounded Degree Vertex Deletion and Defective Coloring are well-studied problems with a rich literature. Bounded Degree Vertex Deletion finds application in a multitude of areas, ranging from computational biology [21] to some related problems in voting theory [7, 9], and its dual problem, called s-Plex Detection, has numerous applications in social network analysis [4, 39, 40]. Various approximation algorithms are known [24, 25, 43]. The problem has also been extensively studied under the scope of parameterized complexity. It is $W[2]$-hard for unbounded values of Δ and parameter k (the value of the optimal) [21], while it admits a linear-size kernel parameterized by k [21, 50], for any fixed $\Delta \geq 0$; numerous FPT algorithms have been presented in the latter setting [40, 41, 49]. FPT approximation algorithms were given for Bounded Degree Vertex Deletion in [34] and [38]. As for Defective Coloring, which also appears in the literature as Improper Coloring, it was introduced almost 40 years ago [1, 13]. The main motivation behind this problem comes from the field of telecommunications, where the colors correspond to available frequencies and the goal is to assign them to communication nodes; a small amount of interference between neighboring nodes may be tolerable, which is modeled by the parameter Δ. There have been plenty of works on the problem (see [2, 3, 5, 6, 12, 30] and the references therein), especially on unit disk graphs and various classes of grids.

The previous work for both problems that is most relevant to us focuses on their parameterized complexity for structural parameters, such as treewidth. In this setting, Bounded Degree Vertex Deletion was one of the first problems to be discovered to be W[1]-hard parameterized by treewidth [8], though the problem does become FPT parameterized by $\mathrm{tw} + \Delta$ or $\mathrm{tw} + k$. This hardness result was more recently improved by Ganian et al. [27], who showed that Bounded Degree Vertex Deletion is W[1]-hard parameterized by tree-depth and feedback vertex set. Defective Coloring was shown to be W[1]-hard parameterized by tree-depth (and hence pathwidth and treewidth) in [5]. However, [5] gave a hardness reduction for pathwidth that is linear in the parameter, and hence implies a $n^{o(\mathrm{pw})}$ lower bound for Defective Coloring under the ETH, but a hardness reduction for tree-depth that is quadratic (implying only a $n^{o(\sqrt{\mathrm{td}})}$ lower bound). Similarly, the reduction given by [27] for Bounded Degree Vertex Deletion parameterized by tree-depth is quartic in the parameter, as it goes through an intermediate problem (a variant of Subset Sum), implying only a $n^{o(\sqrt[4]{\mathrm{td}})}$ lower bound. Defective Coloring is known to be FPT parameterized by vertex cover using a simple win/win argument which applies the treewidth-based DP in one case (if $\Delta > \mathrm{vc}$, then the graph is always 2-colorable; otherwise the standard DP algorithm runs in FPT time), and the same is true for Bounded Degree Vertex Deletion (if $\Delta \leq \mathrm{vc}$, we can use the aforementioned FPT algorithm for parameters $\mathrm{tw} + \Delta$, else assume that $k < \mathrm{vc}$, as otherwise the problem is trivial, follow the reduction of [8] to Vector Dominating Set [44] and notice that at most vc vertices have degree greater than Δ). Hence, the best algorithms for both problems for this parameter have complexity $\mathrm{vc}^{\mathcal{O}(\mathrm{vc})} n^{\mathcal{O}(1)}$.

The fine-grained analysis of the complexity of structural parameterizations, such as by treewidth, is an active field of research. The technique of using the SETH to establish tight running time lower bounds was pioneered by Lokshtanov, Marx, and Saurabh [37]. Since then, tight upper and lower bounds are known for a multitude of problems for parameterizations by treewidth and related parameters, such as pathwidth and clique-width [15, 16, 19, 20, 23, 26, 28, 29, 31, 42, 47]. One difficulty of the results we present here is that

Parameter	Bounded Degree Vertex Deletion	Defective Coloring
pathwidth + Δ	$\mathcal{O}^\star((\Delta+2-\varepsilon)^{\text{pw}})$	$\mathcal{O}^\star((\chi_\text{d}\cdot(\Delta+1)-\varepsilon)^{\text{pw}})$
treedepth	$n^{o(\text{td})}$	$n^{o(\text{td})}$
vertex cover	$\text{vc}^{o(\text{vc})}n^{\mathcal{O}(1)}$	$\text{vc}^{o(\text{vc})}n^{\mathcal{O}(1)}$

Table 1 Lower bounds established in the current work. The results of the first row are under SETH, while all the rest under ETH.

we need to present a family of reductions: one for each fixed value of Δ and χ_d. There are a few other problems for which families of tight lower bounds are known, such as k-Coloring, for which the correct dependence is k^{tw} for treewidth [37] and $(2^k-2)^{\text{cw}}$ for clique-width [35] for all $k \geq 3$; distance r-Dominating Set, for which the correct dependence is $(2r+1)^{\text{tw}}$ [11] and $(3r+1)^{\text{cw}}$ [32], for all $r \geq 1$; and distance d-Independent Set, for which the correct dependence is d^{tw} [33]. In all these cases, the optimal algorithm is the "natural" DP, and our results for Bounded Degree Vertex Deletion and Defective Coloring fit this pattern.

Even though the previous work mentioned above may make it seem that our SETH-based lower bounds are not surprising, it is important to stress that it is not a given that the naïve DP should be optimal for our problems. In particular, Bounded Degree Vertex Deletion falls into a general category of (σ,ρ)-domination problems, which were studied recently in [22] (we refer the reader there for the definition of (σ,ρ)-domination). One of the main results of that work was to show that significant improvements over the basic DP are indeed possible in some cases, and in particular when one of σ,ρ is cofinite. Since Bounded Degree Vertex Deletion is the case where $\sigma = \{0,\dots,\Delta\}$ and $\rho = \mathbb{N}$ (that is, ρ is co-finite), our result falls exactly in the territory left uncharted by [22], where more efficient algorithms could still be found (and where indeed [22] did uncover such algorithms for some values of σ,ρ).

Organization. In Section 2 we discuss the general preliminaries, followed by the lower bounds for Bounded Degree Vertex Deletion in Sections 3–5, and the conclusion in Section 6. Proofs marked with ($\star$), as well as all the results pertaining to Defective Coloring, can be found in the full version of the paper.

2 Preliminaries

Throughout the paper we use standard graph notation [18], and we assume familiarity with the basic notions of parameterized complexity [14]. All graphs considered are undirected without loops, unless explicitly stated otherwise. For a graph $G=(V,E)$ and two integers $\chi_\text{d} \geq 1, \Delta \geq 0$, we say that G admits a (χ_d,Δ)-coloring if one can partition V into χ_d sets such that the graph induced by each set has maximum degree at most Δ. In that case, Defective Coloring is the problem of deciding, given G,χ_d,Δ, whether G admits a (χ_d,Δ)-coloring. For $x,y \in \mathbb{Z}$, let $[x,y] = \{z \in \mathbb{Z} \mid x \leq z \leq y\}$, while $[x] = [1,x]$. Standard $\mathcal{O}^\star$ notation is used to suppress polynomial factors. For function $f : A \to B$, and $A' \subseteq A$, let $f(A') = \sum_{a\in A'} f(a)$. For the pathwidth bounds, we use the notion of *mixed search strategy* [45], where an edge is cleared by either placing a searcher on both of its endpoints or sliding one along the edge. We rely on a weaker form of the ETH, which states that 3-SAT on instances with n variables and m clauses cannot be solved in time $2^{o(n+m)}$.

In k-Multicolored Clique, we are given a graph $G=(V,E)$ and a partition of V into k independent sets $V_1,\dots,V_k$, each of size n, and we are asked to determine whether

G contains a k-clique. It is well-known that this problem does not admit any $f(k)n^{o(k)}$ algorithm, where f is any computable function, unless the ETH is false [14].

In q-CSP-B, we are given a Constraint Satisfaction (CSP) instance with n variables and m constraints. The variables take values in a set Y of size B, i.e. $|Y| = B$. Each constraint involves at most q variables and is given as a list of satisfying assignments for these variables, where a satisfying assignment is a q-tuple of values from the set Y given to each of the q variables. The following result was shown by Lampis [35] to be a natural consequence of the SETH, and has been used in the past for various hardness results [19, 20, 29].

▶ **Theorem 1** ([35]). *For any $B \geq 2$ it holds that, if the SETH is true, then for all $\varepsilon > 0$, there exists a q such that n-variable q-CSP-B cannot be solved in time $\mathcal{O}^\star((B-\varepsilon)^n)$.*

3 Treewidth and Maximum Degree Lower Bounds

In this section we present tight lower bounds on the complexity of solving both Bounded Degree Vertex Deletion and Defective Coloring parameterized by the treewidth of the input graph plus the target degree. The latter result can be found in the full version of the paper.

Both reductions are similar in nature: we start from an instance ϕ of q-CSP-B, and produce an equivalent instance on a graph of pathwidth $\text{pw} = n + \mathcal{O}(1)$, where n denotes the number of variables of ϕ. An interesting observation however, is that for both problems, we have to distinguish between the case where $\Delta = 1$ and $\Delta \geq 2$; the whole construction becomes much more complicated in the second case.

3.1 Bounded Degree Vertex Deletion

In the following, we will present a reduction from q-CSP-B to Bounded Degree Vertex Deletion, for any fixed $\Delta \geq 1$, where $\Delta = B - 2$. In that case, if there exists a $\mathcal{O}^\star((\Delta + 2 - \varepsilon)^{\text{pw}})$ algorithm for Bounded Degree Vertex Deletion, where $\varepsilon > 0$, then there exists a $\mathcal{O}^\star((B-\varepsilon)^n)$ algorithm for q-CSP-B, for any constant q, which due to Theorem 1 results in SETH failing.

Our reduction is based on the construction of "long paths" of *Block gadgets*, that are serially connected in a path-like manner. Each such "path" corresponds to a variable of the given formula, while each column of this construction is associated with one of its constraints. Intuitively, our aim is to embed the B^n possible variable assignments into the $(\Delta + 2)^{\text{tw}}$ states of some optimal dynamic program that would solve the problem on our constructed instance.

Below, we present a sequence of gadgets used in our reduction. The aforementioned block gadgets, which allow a solution to choose among $\Delta + 2$ reasonable choices, are the main ingredient. Notice that these gadgets will differ significantly depending on whether Δ is equal to 1 or not. We connect these gadgets in a path-like manner that ensures that choices remain consistent throughout the construction, and connect constraint gadgets in different "columns" of the constructed grid in a way that allows us to verify if the choice made represents a satisfying assignment, without increasing the graph's treewidth.

▶ **Theorem 2.** *For any constant $\varepsilon > 0$, there is no $\mathcal{O}^\star((3-\varepsilon)^{\text{pw}})$ algorithm deciding Bounded Degree Vertex Deletion for $\Delta = 1$, where* pw *denotes the input graph's pathwidth, unless the SETH is false.*

Proof. Fix some positive $\varepsilon > 0$ for which we want to prove the theorem. We will reduce q-CSP-3, for some q that is a constant that only depends on ε, to Bounded Degree

Vertex Deletion for $\Delta = 1$ in a way that ensures that if the resulting Bounded Degree Vertex Deletion instance could be solved in time $\mathcal{O}^\star((3-\varepsilon)^{\mathrm{pw}})$, then we would obtain an algorithm for q-CSP-3 that would contradict the SETH. To this end, let ϕ be an instance of q-CSP-3 of n variables $X = \{x_i \mid i \in [n]\}$ taking values over the set $Y = [3]$ and m constraints $C = \{c_j \mid j \in [m]\}$. For each constraint we are given a set of at most q variables which are involved in this constraint and a list of satisfying assignments for these variables, the size of which is denoted by $s : C \to [3^q]$, i.e. $s(c_j) \leq 3^q = \mathcal{O}(1)$ denotes the number of satisfying assignments for constraint c_j. We will construct in polynomial time an equivalent instance $\mathcal{I} = (G, k)$ of Bounded Degree Vertex Deletion for $\Delta = 1$, where $\mathrm{pw}(G) \leq n + \mathcal{O}(1)$.

Block and Variable Gadgets. For every variable x_i and every constraint c_j, construct a path of 3 vertices $p^1_{i,j}, p^2_{i,j}$ and $p^3_{i,j}$, which comprises the *block gadget* $\hat{B}_{i,j}$. Intuitively, we will map the deletion of $p^y_{i,j}$ with an assignment where x_i receives value y. Next, for $j \in [m-1]$, we add an edge between $p^3_{i,j}$ and $p^1_{i,j+1}$, thus resulting in n paths $P_1, \dots, P_n$ of length $3m$, called *variable gadgets*.

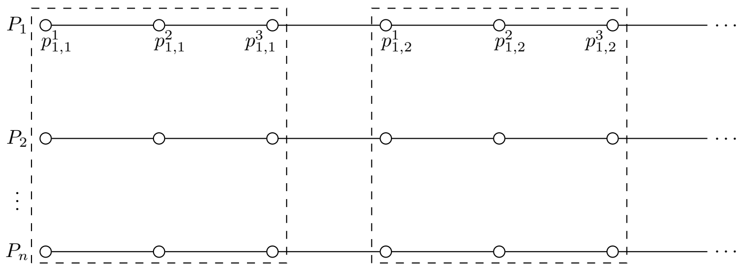


Figure 1 Sequences of block gadgets comprise the variable gadgets.

Constraint Gadget. This gadget is responsible for determining constraint satisfaction, based on the choices made in the rest of the graph. For constraint c_j, construct the *constraint gadget* $\hat{C}_j$ as follows:

- construct a clique of $s(c_j)$ vertices $v^j_1, \dots, v^j_{s(c_j)}$, and fix an arbitrary one-to-one mapping between those vertices and the satisfying assignments of c_j,
- attach to each vertex v^j_ℓ a leaf l^j_ℓ,
- if variable x_i is involved in the constraint c_j and v^j_ℓ corresponds to an assignment where x_i has value $y \in Y$, add an edge between v^j_ℓ and $p^y_{i,j}$.

Let graph G_0 be the graph containing all variable gadgets P_i as well as all the constraint gadgets $\hat{C}_j$, for $i \in [n]$ and $j \in [m]$. To construct graph G, introduce $\kappa = 2n + 1$ copies $G_1, \dots, G_\kappa$ of G_0, such that they are connected sequentially as follows: for $i \in [n]$ and $j \in [\kappa - 1]$, add an edge between $p^3_{i,m}(G_j)$ and $p^1_{i,1}(G_{j+1})$, where $p^y_{i,j}(G_a)$ denotes the vertex $p^y_{i,j}$ of graph G_a. We refer to the block gadget $\hat{B}_{i,j}$, to the variable gadget P_i and to the constraint gadget $\hat{C}_j$ of G_a as $\hat{B}^{G_a}_{i,j}$, $P^{G_a}_i$ and $\hat{C}^{G_a}_j$ respectively. Let $\mathcal{P}^i$ denote the path resulting from $P^{G_1}_i, \dots, P^{G_\kappa}_i$. Let $k' = \sum_{j=1}^m (s(c_j) - 1 + n) = m \cdot n + \sum_{j=1}^m (s(c_j) - 1)$, and set $k = \kappa \cdot k'$.

▶ **Lemma 3.** ($\star$) *If ϕ is satisfiable, then there exists $S \subseteq V(G)$ such that $G - S$ has maximum degree at most 1 and $|S| \leq k$.*

▶ **Lemma 4.** ($\star$) *If there exists $S \subseteq V(G)$ such that $G - S$ has maximum degree at most 1 and $|S| \leq k$, then ϕ is satisfiable.*

▶ **Lemma 5.** ($\star$) *It holds that* $\text{pw}(G) \leq n + \mathcal{O}(1)$*.*

Therefore, in polynomial time, we can construct a graph G, of pathwidth $\text{pw}(G) \leq n + \mathcal{O}(1)$ due to Lemma 5, such that, due to Lemmas 3 and 4, deciding whether there exists $S \subseteq V(G)$ of size $|S| \leq k$ and $G - S$ has maximum degree at most 1 is equivalent to deciding whether ϕ is satisfiable. In that case, assuming there exists a $\mathcal{O}^\star((3-\varepsilon)^{\text{pw}(G)})$ algorithm for Bounded Degree Vertex Deletion for $\Delta = 1$, one could decide q-CSP-3 in time $\mathcal{O}^\star((3-\varepsilon)^{\text{pw}(G)}) = \mathcal{O}^\star((3-\varepsilon)^{n+\mathcal{O}(1)}) = \mathcal{O}^\star((3-\varepsilon)^n)$ for any constant q, which contradicts the SETH due to Theorem 1. ◀

▶ **Theorem 6.** ($\star$)*For any constant* $\varepsilon > 0$*, there is no* $\mathcal{O}^\star((\Delta+2-\varepsilon)^{\text{pw}})$ *algorithm deciding* Bounded Degree Vertex Deletion *for* $\Delta \geq 2$*, where* pw *denotes the input graph's pathwidth, unless the SETH is false.*

4 Tree-depth Lower Bounds

In this section we present lower bounds on the complexity of solving Bounded Degree Vertex Deletion and Defective Coloring, when parameterized by the tree-depth of the input graph. The latter result can be found in the full version of the paper.

The common starting point of both reductions is an instance of k-Multicolored Clique, where k is a power of 2. Our main technical contribution is a recursive construction which allows us to keep the tree-depth of the constructed graph linear with respect to k, which we briefly sketch here. The main idea behind the construction is the following:

- First, we describe a *choice gadget* $\hat{C}_i$, which encodes, for every independent set V_i of the graph, which vertex of V_i is part of the clique.
- Afterwards, we describe how one can make a *copy* of such a choice gadget, by using only a constant number of vertices, while at the same time guaranteeing that the choices between the instances of the choice gadget remain the same.
- Lastly, we define an *adjacency gadget* $\hat{A}(i_1, i_2, i'_1, i'_2)$, whose purpose is to verify that, for the given choices of vertices, there exists an edge between V_i and $V_{i'}$, for any $i \in [i_1, i_2]$ and $i' \in [i'_1, i'_2]$. Initially we deal with the case where $i_1 = i_2$ and $i'_1 = i'_2$, while ensuring that the tree-depth of the construction is constant. For the other case, the gadget is constructed in two steps. Firstly, it contains all the choice gadgets $\hat{C}_i$ and $\hat{C}_{i'}$. Secondly, it contains 4 instances of adjacency gadgets, due to the upper and lower half of $[i_1, i_2]$ and $[i'_1, i'_2]$, while the occurrences of the choice gadgets in those are copies of the choice gadgets introduced in the first step. The fact that k is a power of 2 guarantees that the upper and lower half of both $[i_1, i_2]$ and $[i'_1, i'_2]$ are well defined.

Then, by removing the vertices used in the copy gadgets, it follows that all adjacency gadgets constructed in the second step become disconnected. Therefore, the tree-depth of the whole construction is given by a recursive formula of the form $T(k) = \mathcal{O}(k) + T(k/2)$.

4.1 Bounded Degree Vertex Deletion

▶ **Theorem 7.** *For any computable function* f*, if there exists an algorithm that solves* Bounded Degree Vertex Deletion *in time* $f(\text{td})n^{o(\text{td})}$*, where* td *denotes the tree-depth of the input graph, then the ETH is false.*

Proof. Let (G, k) be an instance of k-Multicolored Clique, such that every vertex of G has a self loop, i.e. $\{v, v\} \in E(G)$, for all $v \in V(G)$. Recall that we assume that G is given to us partitioned into k independent sets $V_1, \dots, V_k$, where $V_i = \{v^i_1, \dots, v^i_n\}$. Assume without

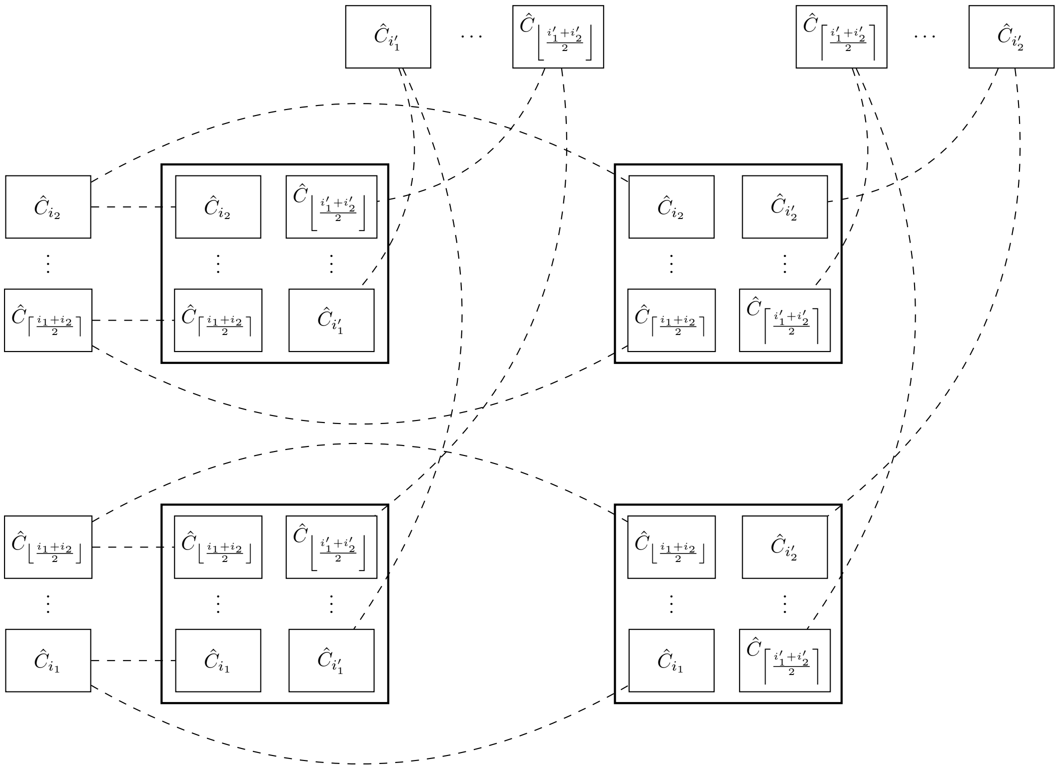


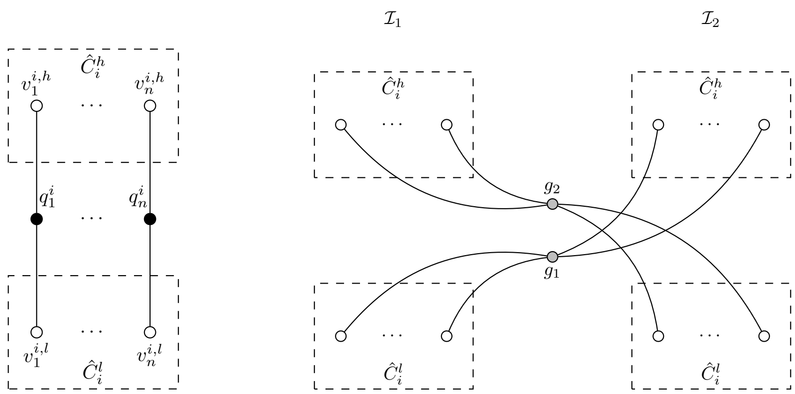
Figure 2 Adjacency gadget $\hat{A}(i_1, i_2, i'_1, i'_2)$. Dashed lines denote copies.

loss of generality that $k = 2^z$, for some $z \in \mathbb{N}$ (one can do so by adding dummy independent sets connected to all the other vertices of the graph). Moreover, let $E^{i_1,i_2} \subseteq E(G)$ denote the edges of G with one endpoint in V_{i_1} and the other in V_{i_2}. Set $\Delta = n^3$. We will construct in polynomial time a graph H of tree-depth $\text{td}(H) = \mathcal{O}(k)$ and size $|V(H)| = k^{\mathcal{O}(1)} \cdot n^{\mathcal{O}(1)}$, such that there exists $S \subseteq V(H)$, $|S| \leq k'$ and $H - S$ has maximum degree at most Δ, for some k', if and only if G has a k-clique.

Choice Gadget. For an independent set V_i, we construct the *choice gadget* $\hat{C}_i$ as depicted in Figure 3a. We first construct independent sets $\hat{C}_i^p = \{v_1^{i,p}, \dots, v_n^{i,p}\}$, where $p \in \{h, l\}$. Afterwards, we connect $v_j^{i,h}$ and $v_j^{i,l}$ with a vertex q_j^i, and add to the latter $\Delta - 1$ leaves. Intuitively, we will consider an one-to-one mapping between the vertex v_j^i of V_i belonging to a supposed k-clique of G and the deletion of exactly j vertices of $\hat{C}_i^l$ and $n - j$ from $\hat{C}_i^h$.

Copy Gadget. Given two instances $\mathcal{I}_1$, $\mathcal{I}_2$ of a choice gadget $\hat{C}_i$, when we say that we connect them with a *copy gadget*, we introduce two vertices g_1 and g_2, attach to each of those $\Delta - n$ leaves, and lastly add an edge between g_1 (respectively, g_2) with the vertices of $\hat{C}_i^l$ of instance $\mathcal{I}_1$ (respectively, $\mathcal{I}_2$), as well as the vertices of $\hat{C}_i^h$ of instance $\mathcal{I}_2$ (respectively, $\mathcal{I}_1$).

Edge Gadget. Let $e = \{v_{j_1}^{i_1}, v_{j_2}^{i_2}\} \in E^{i_1,i_2}$ be an edge of G. Construct the *edge gadget* $\hat{E}_e$ as depicted in Figure 4, where every vertex c_j^i has Δ leaves attached.



(a) Choice gadget $\hat{C}_i$. **(b)** Making a copy of a choice gadget $\hat{C}_i$.

Figure 3 Black vertices have $\Delta - 1$ and gray $\Delta - n$ leaves attached.

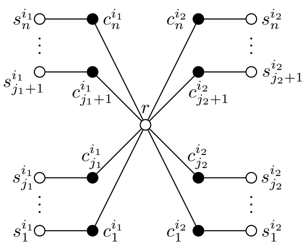


Figure 4 Edge gadget $\hat{E}_e$ for $e = \{v_{j_1}^{i_1}, v_{j_2}^{i_2}\}$. Black vertices have Δ leaves attached.

Adjacency Gadget. For $i_1 \leq i_2$ and $i'_1 \leq i'_2$, we define the *adjacency gadget* $\hat{A}(i_1, i_2, i'_1, i'_2)$ as follows:

- Consider first the case when $i_1 = i_2$ and $i'_1 = i'_2$. Let the adjacency gadget contain instances of the edge gadgets $\hat{E}_e$, for $e \in E^{i_1,i'_1}$, the choice gadgets $\hat{C}_{i_1}$ and $\hat{C}_{i'_1}$, as well as vertices $\ell^l_{i_1,i'_1}$, $\ell^h_{i_1,i'_1}$, $r^l_{i_1,i'_1}$ and $r^h_{i_1,i'_1}$. Add edges between

  - $\ell^l_{i_1,i'_1}$ and $\hat{C}^l_{i_1}$,
  - $\ell^h_{i_1,i'_1}$ and $\hat{C}^h_{i_1}$,
  - $r^l_{i_1,i'_1}$ and $\hat{C}^l_{i'_1}$,
  - $r^h_{i_1,i'_1}$ and $\hat{C}^h_{i'_1}$.

  If $e = \{v_{j_1}^{i_1}, v_{j_2}^{i'_1}\} \in E^{i_1,i'_1}$, then add the following edges adjacent to $\hat{E}_e$:

  - $\ell^l_{i_1,i'_1}$ with $s_\kappa^{i_1}$, for $\kappa \in [j_1]$,
  - $\ell^h_{i_1,i'_1}$ with $s_\kappa^{i_1}$, for $\kappa \in [j_1 + 1, n]$,
  - $r^l_{i_1,i'_1}$ with $s_\kappa^{i'_1}$, for $\kappa \in [j_2]$,
  - $r^h_{i_1,i'_1}$ with $s_\kappa^{i'_1}$, for $\kappa \in [j_2 + 1, n]$.

  Let $\tau(x)$, where $x \in \{\ell^l_{i_1,i'_1}, \ell^h_{i_1,i'_1}, r^l_{i_1,i'_1}, r^h_{i_1,i'_1}\}$, denote the number of neighbors of x belonging to some edge gadget. Attach $\Delta - \tau(x)$ leaves to vertex x.
- Now consider the case when $i_1 < i_2$ and $i'_1 < i'_2$. Then, let $\hat{A}(i_1, i_2, i'_1, i'_2)$ contain choice gadgets $\hat{C}_i$ and $\hat{C}_{i'}$, where $i \in [i_1, i_2]$ and $i' \in [i'_1, i'_2]$, which we will refer to as the *original choice gadgets* of $\hat{A}(i_1, i_2, i'_1, i'_2)$, as well as the adjacency gadgets

  - $\hat{A}(i_1, \lfloor \frac{i_1+i_2}{2} \rfloor, i'_1, \left\lfloor \frac{i'_1+i'_2}{2} \right\rfloor)$,
  - $\hat{A}(i_1, \lfloor \frac{i_1+i_2}{2} \rfloor, \left\lceil \frac{i'_1+i'_2}{2} \right\rceil, i'_2)$,
  - $\hat{A}(\lceil \frac{i_1+i_2}{2} \rceil, i_2, i'_1, \left\lfloor \frac{i'_1+i'_2}{2} \right\rfloor)$,
  - $\hat{A}(\lceil \frac{i_1+i_2}{2} \rceil, i_2, \left\lceil \frac{i'_1+i'_2}{2} \right\rceil, i'_2)$.

Lastly, we connect with a copy gadget any choice gadgets $\hat{C}_i$ and $\hat{C}_{i'}$ appearing in said adjacency gadgets, with the corresponding original choice gadget $\hat{C}_i$ and $\hat{C}_{i'}$. Notice that then, every original choice gadget is taking part in two copy gadgets.

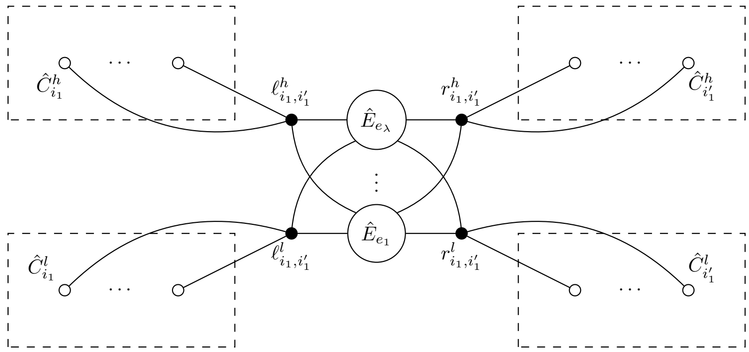


Figure 5 Adjacency gadget $\hat{A}(i_1, i_1, i'_1, i'_1)$, where $E^{i_1,i'_1} = \{e_i \mid i \in [\lambda]\}$. Black vertices have leaves attached.

Let graph H be the adjacency gadget $\hat{A}(1, k, 1, k)$. Notice that it holds that $|V(H)| = (n \cdot k)^{\mathcal{O}(1)}$. Let $\beta = 2k(2k-1)$, and set $k' = 2(|E(G)| - kn) \cdot 2n + kn \cdot 2n + 2\binom{k}{2} + k + n \cdot \beta$.

▶ **Lemma 8.** (⋆) *H has the following properties:*

- *The number of instances of choice gadgets present in H is β,*
- *The number of instances of edge gadget $\hat{E}_e$ present in H, where $e = \{v^{i_1}_{j_1}, v^{i_2}_{j_2}\} \in E(G)$, is one if $i_1 = i_2$, and two otherwise.*

▶ **Lemma 9.** (⋆) *It holds that $\mathrm{td}(H) = \mathcal{O}(k)$.*

▶ **Lemma 10.** (⋆) *If G contains a k-clique, then there exists $S \subseteq V(H)$, with $|S| \le k'$, such that $H - S$ has maximum degree at most Δ.*

▶ **Lemma 11.** (⋆) *If there exists $S \subseteq V(H)$, with $|S| \le k'$, such that $H - S$ has maximum degree at most Δ, then G contains a k-clique.*

Therefore, in polynomial time, we can construct a graph H, of tree-depth $\mathrm{td} = \mathcal{O}(k)$ due to Lemma 9, such that, due to Lemmas 10 and 11, deciding whether there exists $S \subseteq V(H)$ of size $|S| \le k'$ and $H - S$ has maximum degree at most $\Delta = n^3$ is equivalent to deciding whether G has a k-clique. In that case, assuming there exists a $f(\mathrm{td})|V(H)|^{o(\mathrm{td})}$ algorithm for Bounded Degree Vertex Deletion, where f is any computable function, one could decide k-Multicolored Clique in time $f(\mathrm{td})|V(H)|^{o(\mathrm{td})} = g(k) \cdot n^{o(k)}$, for some computable function g, which contradicts the ETH. ◀

5 Vertex Cover Lower Bounds

In this section we present lower bounds on the complexity of solving Bounded Degree Vertex Deletion when parameterized by the vertex cover of the input graph. An analogous lower bound is shown for Defective Coloring, which has been deferred to the appendix due to space restrictions. In both cases, we start from a 3-SAT instance of n variables, and produce an equivalent instance where the input graph has vertex cover $\mathcal{O}(n/\log n)$, hence any algorithm solving the latter problem in time $\mathrm{vc}^{o(\mathrm{vc})} n^{\mathcal{O}(1)}$ would refute the ETH. As a consequence of the above, already known algorithms for both of these problems are essentially optimal. We start by presenting some necessary tools used in both of these reductions, and then prove the stated results for Bounded Degree Vertex Deletion.

5.1 Preliminary Tools

We first define a constrained version of 3-SAT, called (3,4)-XSAT. This variant is closely related with the (3,4)-SAT problem [46] which asks whether a given formula ϕ is satisfiable, where ϕ is a 3-SAT formula each clause of which contains exactly 3 different variables and each variable occurs in at most 4 clauses. As observed by Bonamy et al. [10], a corollary of Tovey's work [46] is that there is no $2^{o(n)}$ algorithm for (3,4)-SAT unless the ETH is false, where n denotes the number of variables of the formula. Here we prove an analogous lower bound for (3,4)-XSAT. Subsequently, by closely following Lemma 3.2 from [10], we present a way to partition the formula's variables and clauses into groups such that variables appearing in clauses of the same clause group belong to different variable groups.

(3,4)-XSAT
Input: A 3-SAT formula ϕ every clause of which contains exactly 3 distinct variables and each variable appears in at most 4 clauses.
Task: Determine whether there exists an assignment to the variables of ϕ such that each clause has exactly one True literal.

▶ **Theorem 12.** ($\star$) *(3,4)-XSAT cannot be decided in time $2^{o(n)}$, where n denotes the number of variables of the input formula, unless the ETH fails.*

We proceed by proving that, given a (3,4)-XSAT instance, we can partition the variables and clauses of the formula into groups such that variables appearing in clauses of the same clause group belong to different variable groups.

▶ **Lemma 13.** ($\star$) *Let ϕ be an instance of (3,4)-XSAT, where V denotes the set of its n variables and C the set of its clauses. Moreover, let $b \leq \sqrt{n}$. One can produce in time $n^{\mathcal{O}(1)}$ a partition of ϕ's variables into n_V disjoint sets $V_1, \dots, V_{n_V}$ of size at most b as well as a partition of its clauses into n_C disjoint sets $C_1, \dots, C_{n_C}$ of size at most $\sqrt{n}$, for some integers $n_V = \mathcal{O}(n/b)$ and $n_C = \mathcal{O}(\sqrt{n})$, such that, for any $i \in [n_C]$, any two variables appearing in clauses of C_i belong to different variable subsets.*

▶ **Definition 14.** *A d-detecting family is a set of subsets of a finite set U that can be used to distinguish between different functions $f, g : U \to \{0, \dots, d-1\}$. Therefore, if $f \neq g$, there exists $U' \subseteq U$ such that $f(U') \neq g(U')$ and U' belongs to said family.*

Lindström [36] has provided a deterministic construction of sublinear, d-detecting families, while Bonamy et al. [10] were the first to use them in the context of computational complexity, proving tight lower bounds for the Multicoloring problem under the ETH. The following theorem will be crucial towards proving the stated lower bounds.

▶ **Theorem 15** ([36])**.** *For every constant $d \in \mathbb{N}$ and finite set U, there is a d-detecting family $\mathcal{F}$ on U of size $\frac{2|U|}{\log_d |U|} \cdot (1 + o(1))$. Moreover, $\mathcal{F}$ can be constructed in time polynomial in $|U|$.*

5.2 Bounded Degree Vertex Deletion

Let ϕ be an instance of (3,4)-XSAT of n variables. Assume without loss of generality that n is a power of 4 (this can be achieved by adding dummy variables to the instance if needed). Making use of Lemma 13, one can obtain in time $n^{\mathcal{O}(1)}$ the following:

- a partition of ϕ's variables into subsets $V_1, \ldots, V_{n_V}$, where $|V_i| \leq \log n$ and $n_V = \mathcal{O}(n/\log n)$,
- a partition of ϕ's clauses into subsets $C_1, \ldots, C_{n_C}$, where $|C_i| \leq \sqrt{n}$ and $n_C = \mathcal{O}(\sqrt{n})$,

where any two variables occurring in clauses of the same clause subset belong to different variable subsets. For $i \in [n_C]$, let $\{C_{i,1}, \ldots, C_{i,n^i_{\mathcal{F}}}\}$ be a 4-detecting family of subsets of C_i for some $n^i_{\mathcal{F}} = \mathcal{O}(\sqrt{n}/\log n)$, produced in time $n^{\mathcal{O}(1)}$ due to Theorem 15. Moreover, let $n_{\mathcal{F}} = \max_{i=1}^{n_C} n^i_{\mathcal{F}}$. Define $\Delta = n^3$ and $k = n_V$. We will construct a graph $G = (V, E)$ such that there exists $S \subseteq V(G)$ of size $|S| \leq k$ and $G - S$ has maximum degree at most Δ if and only if there exists an assignment such that every clause of ϕ has exactly one True literal.

Choice Gadget. For each variable subset V_i, we define the choice gadget graph G_i as follows:

- introduce vertices κ_i, λ_i and v_i^j, where $j \in [n]$,
- add edges $\{\kappa_i, v_i^j\}$ and $\{\lambda_i, v_i^j\}$, for all $j \in [n]$,
- attach sufficiently many leaves to κ_i and λ_i such that their degree is $\Delta + 1$.

Let $\mathcal{V}_i = \{v_i^j \mid j \in [n]\}$, for $i = 1, \ldots, n_V$. We fix an arbitrary one-to-one mapping so that every vertex of $\mathcal{V}_i$ corresponds to a different assignment for the variables of V_i. Since $2^{|V_i|} \leq n$, there are sufficiently many vertices to uniquely encode all the different assignments of V_i. Let $\mathcal{V} = \mathcal{V}_1 \cup \ldots \cup \mathcal{V}_{n_V}$ denote the set of all such vertices.

Clause Gadget. For $i \in [n_C]$, let C_i be a clause subset and $\{C_{i,1}, \ldots, C_{i,n^i_{\mathcal{F}}}\}$ its 4-detecting family. For every subset $C_{i,j}$ of the 4-detecting family, introduce vertices $c_{i,j}$ and $c'_{i,j}$. Add an edge between $c_{i,j}$ and v_p^q if there exists variable $x \in V_p$ such that x occurs in some clause $c \in C_{i,j}$, and v_p^q corresponds to an assignment of V_p that satisfies c. Due to Lemma 13, $c_{i,j}$ has exactly $|C_{i,j}| \cdot \frac{3n}{2}$ such edges: there are exactly $3|C_{i,j}|$ different variables appearing in clauses of $C_{i,j}$, each belonging to a different variable subset, and for each such variable, half the assignments of the corresponding variable subset result in the satisfaction of the corresponding clause of $C_{i,j}$. Attach to $c_{i,j}$ a sufficient number of leaves such that its total degree is $\Delta + |C_{i,j}|$. Moreover, for $v \in \mathcal{V}$, let $v \in N(c'_{i,j})$ if $v \notin N(c_{i,j})$. Notice that then, it holds that $N(c_{i,j}) \cup N(c'_{i,j}) \supseteq \mathcal{V}$, while $N(c_{i,j}) \cap N(c'_{i,j}) = \emptyset$. Lastly, attach to $c'_{i,j}$ a sufficient number of leaves such that its total degree is $\Delta + (k - |C_{i,j}|)$.

Let $\mathcal{I} = (G, \Delta, k)$ be an instance of Bounded Degree Vertex Deletion.

▶ **Lemma 16.** ($\star$)*It holds that* $\text{vc}(G) = \mathcal{O}(n/\log n)$.

▶ **Lemma 17.** ($\star$) *If ϕ is a Yes instance of (3,4)-XSAT, then $\mathcal{I}$ is a Yes instance of Bounded Degree Vertex Deletion.*

▶ **Lemma 18.** ($\star$) *If $\mathcal{I}$ is a Yes instance of Bounded Degree Vertex Deletion, then ϕ is a Yes instance of (3,4)-XSAT.*

We can now prove the main theorem of this section.

▶ **Theorem 19.** ($\star$) *There is no* $\text{vc}^{o(\text{vc})} n^{\mathcal{O}(1)}$ *time algorithm for Bounded Degree Vertex Deletion, where* vc *denotes the size of the minimum vertex cover of the input graph, unless the ETH fails.*

6 Conclusion

In this work, we have examined in depth the complexity of Bounded Degree Vertex Deletion and Defective Coloring under the perspective of parameterized complexity.

In particular, we have precisely determined the complexity of both problems parameterized by some of the most commonly used structural parameters. As a direction for future research, we consider the question of whether we could obtain a $n^{o(\mathrm{fvs})}$ lower bound for Bounded Degree Vertex Deletion as well as for Defective Coloring when $\chi_{\mathrm{d}} = 2$, where fvs denotes the size of the minimum feedback vertex set of the input graph.

References

1 James A. Andrews and Michael S. Jacobson. On a generalization of chromatic number. *Congressus Numerantium*, 47:33–48, 1985.

2 Patrizio Angelini, Michael A. Bekos, Felice De Luca, Walter Didimo, Michael Kaufmann, Stephen G. Kobourov, Fabrizio Montecchiani, Chrysanthi N. Raftopoulou, Vincenzo Roselli, and Antonios Symvonis. Vertex-coloring with defects. *J. Graph Algorithms Appl.*, 21(3):313–340, 2017. doi:10.7155/jgaa.00418.

3 Dan Archdeacon. A note on defective colorings of graphs in surfaces. *J. Graph Theory*, 11(4):517–519, 1987. doi:10.1002/jgt.3190110408.

4 Balabhaskar Balasundaram, Sergiy Butenko, and Illya V. Hicks. Clique relaxations in social network analysis: The maximum *k*-plex problem. *Oper. Res.*, 59(1):133–142, 2011. doi:10.1287/opre.1100.0851.

5 Rémy Belmonte, Michael Lampis, and Valia Mitsou. Parameterized (approximate) defective coloring. *SIAM J. Discret. Math.*, 34(2):1084–1106, 2020. doi:10.1137/18M1223666.

6 Rémy Belmonte, Michael Lampis, and Valia Mitsou. Defective coloring on classes of perfect graphs. *Discret. Math. Theor. Comput. Sci.*, 24, 2022. doi:10.46298/dmtcs.4926.

7 Nadja Betzler, Hans L. Bodlaender, Robert Bredereck, Rolf Niedermeier, and Johannes Uhlmann. On making a distinguished vertex of minimum degree by vertex deletion. *Algorithmica*, 68(3):715–738, 2014. doi:10.1007/s00453-012-9695-6.

8 Nadja Betzler, Robert Bredereck, Rolf Niedermeier, and Johannes Uhlmann. On bounded-degree vertex deletion parameterized by treewidth. *Discret. Appl. Math.*, 160(1-2):53–60, 2012. doi:10.1016/j.dam.2011.08.013.

9 Nadja Betzler and Johannes Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theor. Comput. Sci.*, 410(52):5425–5442, 2009. doi:10.1016/j.tcs.2009.05.029.

10 Marthe Bonamy, Lukasz Kowalik, Michal Pilipczuk, Arkadiusz Socala, and Marcin Wrochna. Tight lower bounds for the complexity of multicoloring. *ACM Trans. Comput. Theory*, 11(3):13:1–13:19, 2019. doi:10.1145/3313906.

11 Glencora Borradaile and Hung Le. Optimal dynamic program for r-domination problems over tree decompositions. In *11th International Symposium on Parameterized and Exact Computation, IPEC 2016*, volume 63 of *LIPIcs*, pages 8:1–8:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.IPEC.2016.8.

12 Ilkyoo Choi and Louis Esperet. Improper coloring of graphs on surfaces. *J. Graph Theory*, 91(1):16–34, 2019. doi:10.1002/jgt.22418.

13 Lenore J. Cowen, Robert Cowen, and Douglas R. Woodall. Defective colorings of graphs in surfaces: Partitions into subgraphs of bounded valency. *J. Graph Theory*, 10(2):187–195, 1986. doi:10.1002/jgt.3190100207.

14 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms.* Springer, 2015. doi:10.1007/978-3-319-21275-3.

15 Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Fast hamiltonicity checking via bases of perfect matchings. *J. ACM*, 65(3):12:1–12:46, 2018. doi:10.1145/3148227.

16 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. *ACM Trans. Algorithms*, 18(2):17:1–17:31, 2022. doi:10.1145/3506707.

17 Marek Cygan and Marcin Pilipczuk. Exact and approximate bandwidth. *Theor. Comput. Sci.*, 411(40-42):3701–3713, 2010. doi:10.1016/j.tcs.2010.06.018.

18 Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate texts in mathematics*. Springer, 2017. doi:10.1007/978-3-662-53622-3.

19 Louis Dublois, Michael Lampis, and Vangelis Th. Paschos. New algorithms for mixed dominating set. *Discret. Math. Theor. Comput. Sci.*, 23(1), 2021. doi:10.46298/dmtcs.6824.

20 Louis Dublois, Michael Lampis, and Vangelis Th. Paschos. Upper dominating set: Tight algorithms for pathwidth and sub-exponential approximation. *Theor. Comput. Sci.*, 923:271–291, 2022. doi:10.1016/j.tcs.2022.05.013.

21 Michael R. Fellows, Jiong Guo, Hannes Moser, and Rolf Niedermeier. A generalization of nemhauser and trotter's local optimization theorem. *J. Comput. Syst. Sci.*, 77(6):1141–1158, 2011. doi:10.1016/j.jcss.2010.12.001.

22 Jacob Focke, Dániel Marx, Fionn Mc Inerney, Daniel Neuen, Govind S. Sankar, Philipp Schepper, and Philip Wellnitz. Tight complexity bounds for counting generalized dominating sets in bounded-treewidth graphs. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023*, pages 3664–3683. SIAM, 2023. doi:10.1137/1.9781611977554.ch140.

23 Jacob Focke, Dániel Marx, and Pawel Rzazewski. Counting list homomorphisms from graphs of bounded treewidth: tight complexity bounds. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022*, pages 431–458. SIAM, 2022. doi:10.1137/1.9781611977073.22.

24 Toshihiro Fujito. A unified approximation algorithm for node-deletion problems. *Discret. Appl. Math.*, 86(2-3):213–231, 1998. doi:10.1016/S0166-218X(98)00035-3.

25 Toshihiro Fujito. Approximating bounded degree deletion via matroid matching. In *Algorithms and Complexity - 10th International Conference, CIAC 2017*, volume 10236 of *Lecture Notes in Computer Science*, pages 234–246, 2017. doi:10.1007/978-3-319-57586-5_20.

26 Robert Ganian, Thekla Hamm, Viktoriia Korchemna, Karolina Okrasa, and Kirill Simonov. The fine-grained complexity of graph homomorphism parameterized by clique-width. In *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022*, volume 229 of *LIPIcs*, pages 66:1–66:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.66.

27 Robert Ganian, Fabian Klute, and Sebastian Ordyniak. On structural parameterizations of the bounded-degree vertex deletion problem. *Algorithmica*, 83(1):297–336, 2021. doi:10.1007/s00453-020-00758-8.

28 Carla Groenland, Isja Mannens, Jesper Nederlof, and Krisztina Szilágyi. Tight bounds for counting colorings and connected edge sets parameterized by cutwidth. In *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022*, volume 219 of *LIPIcs*, pages 36:1–36:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.STACS.2022.36.

29 Tesshu Hanaka, Ioannis Katsikarelis, Michael Lampis, Yota Otachi, and Florian Sikora. Parameterized orientable deletion. *Algorithmica*, 82(7):1909–1938, 2020. doi:10.1007/s00453-020-00679-6.

30 Frédéric Havet, Ross J. Kang, and Jean-Sébastien Sereni. Improper coloring of unit disk graphs. *Networks*, 54(3):150–164, 2009. doi:10.1002/net.20318.

31 Lars Jaffke and Bart M. P. Jansen. Fine-grained parameterized complexity analysis of graph coloring problems. *Discret. Appl. Math.*, 327:33–46, 2023. doi:10.1016/j.dam.2022.11.011.

32 Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Structural parameters, tight bounds, and approximation for (k, r)-center. *Discret. Appl. Math.*, 264:90–117, 2019. doi:10.1016/j.dam.2018.11.002.

33 Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Structurally parameterized d-scattered set. *Discret. Appl. Math.*, 308:168–186, 2022. doi:10.1016/j.dam.2020.03.052.

34 Michael Lampis. Parameterized approximation schemes using graph widths. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014*, volume 8572 of *Lecture Notes in Computer Science*, pages 775–786. Springer, 2014. doi:10.1007/978-3-662-43948-7_64.

35 Michael Lampis. Finer tight bounds for coloring on clique-width. *SIAM J. Discret. Math.*, 34(3):1538–1558, 2020. doi:10.1137/19M1280326.

36 Bernt Lindström. On a combinatorial problem in number theory. *Canadian Mathematical Bulletin*, 8(4):477–490, 1965. doi:10.4153/CMB-1965-034-2.

37 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, 2018. doi:10.1145/3170442.

38 Daniel Lokshtanov, Pranabendu Misra, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Fpt-approximation for FPT problems. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021*, pages 199–218. SIAM, 2021. doi:10.1137/1.9781611976465.14.

39 Benjamin McClosky and Illya V. Hicks. Combinatorial algorithms for the maximum k-plex problem. *J. Comb. Optim.*, 23(1):29–49, 2012. doi:10.1007/s10878-010-9338-2.

40 Hannes Moser, Rolf Niedermeier, and Manuel Sorge. Exact combinatorial algorithms and experiments for finding maximum k-plexes. *J. Comb. Optim.*, 24(3):347–373, 2012. doi:10.1007/s10878-011-9391-5.

41 Naomi Nishimura, Prabhakar Ragde, and Dimitrios M. Thilikos. Fast fixed-parameter tractable algorithms for nontrivial generalizations of vertex cover. *Discret. Appl. Math.*, 152(1-3):229–245, 2005. doi:10.1016/j.dam.2005.02.029.

42 Karolina Okrasa and Pawel Rzazewski. Fine-grained complexity of the graph homomorphism problem for bounded-treewidth graphs. *SIAM J. Comput.*, 50(2):487–508, 2021. doi:10.1137/20M1320146.

43 Michael Okun and Amnon Barak. A new approach for approximating node deletion problems. *Inf. Process. Lett.*, 88(5):231–236, 2003. doi:10.1016/j.ipl.2003.08.005.

44 Venkatesh Raman, Saket Saurabh, and Sriganesh Srihari. Parameterized algorithms for generalized domination. In *Combinatorial Optimization and Applications, Second International Conference, COCOA 2008*, volume 5165 of *Lecture Notes in Computer Science*, pages 116–126. Springer, 2008. doi:10.1007/978-3-540-85097-7_11.

45 Atsushi Takahashi, Shuichi Ueno, and Yoji Kajitani. Mixed searching and proper-path-width. *Theor. Comput. Sci.*, 137(2):253–268, 1995.

46 Craig A. Tovey. A simplified np-complete satisfiability problem. *Discret. Appl. Math.*, 8(1):85–89, 1984. doi:10.1016/0166-218X(84)90081-7.

47 Bas A. M. van Geffen, Bart M. P. Jansen, Arnoud A. W. M. de Kroon, and Rolf Morel. Lower bounds for dynamic programming on planar graphs of bounded cutwidth. *J. Graph Algorithms Appl.*, 24(3):461–482, 2020. doi:10.7155/jgaa.00542.

48 Johan M. M. van Rooij. A generic convolution algorithm for join operations on tree decompositions. In *Computer Science - Theory and Applications - 16th International Computer Science Symposium in Russia, CSR 2021*, volume 12730 of *Lecture Notes in Computer Science*, pages 435–459. Springer, 2021. doi:10.1007/978-3-030-79416-3_27.

49 Mingyu Xiao. A parameterized algorithm for bounded-degree vertex deletion. In *Computing and Combinatorics - 22nd International Conference, COCOON 2016*, volume 9797 of *Lecture Notes in Computer Science*, pages 79–91. Springer, 2016. doi:10.1007/978-3-319-42634-1_7.

50 Mingyu Xiao. On a generalization of nemhauser and trotter's local optimization theorem. *J. Comput. Syst. Sci.*, 84:97–106, 2017. doi:10.1016/j.jcss.2016.08.003.