Towards Exact Structural Thresholds for Parameterized Complexity

Falko Hegerfeld
Humboldt-Universität zu Berlin, Germany

Stefan Kratsch
Humboldt-Universität zu Berlin, Germany

Abstract

Parameterized complexity seeks to optimally use input structure to obtain faster algorithms for NP-hard problems. This has been most successful for graphs of low treewidth, i.e., graphs decomposable by small separators: Many problems admit fast algorithms relative to treewidth and many of them are optimal under the Strong Exponential-Time Hypothesis (SETH). Fewer such results are known for more general structure such as low clique-width (decomposition by large and dense but structured separators) and more restrictive structure such as low deletion distance to some sparse graph class.

Despite these successes, such results remain "islands" within the realm of possible structure. Rather than adding more islands, we seek to determine the transitions between them, that is, we aim for structural thresholds where the complexity increases as input structure becomes more general. Going from deletion distance to treewidth, is a single deletion set to a graph with simple components enough to yield the same lower bound as for treewidth or does it take many disjoint separators? Going from treewidth to clique-width, how much more density entails the same complexity as clique-width? Conversely, what is the most restrictive structure that yields the same lower bound?

For treewidth, we obtain both refined and new lower bounds that apply already to graphs with a single separator X such that $G - X$ has treewidth at most $r = \mathcal{O}(1)$, while G has treewidth $|X| + \mathcal{O}(1)$. We rule out algorithms running in time $\mathcal{O}^*((r+1-\varepsilon)^k)$ for Deletion to r-Colorable parameterized by $k = |X|$; this implies the same lower bound relative to treedepth and (hence) also to treewidth. It specializes to $\mathcal{O}^*((3-\varepsilon)^k)$ for Odd Cycle Transversal where $\mathrm{tw}(G - X) \leq r = 2$ is best possible. For clique-width, an extended version of the above reduction rules out time $\mathcal{O}^*((4-\varepsilon)^k)$, where X is allowed to be a possibly large separator consisting of k (true) twinclasses, while the treewidth of $G - X$ remains r; this is proved also for the more general Deletion to r-Colorable and it implies the same lower bound relative to clique-width. Further results complement what is known for Vertex Cover, Dominating Set and Maximum Cut. All lower bounds are matched by existing and newly designed algorithms.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Mathematics of computing → Graph algorithms

Keywords and phrases Parameterized complexity, lower bound, vertex cover, odd cycle transversal, SETH, modulator, treedepth, cliquewidth

Digital Object Identifier 10.4230/LIPIcs.IPEC.2022.17

Related Version *Full Version*: https://arxiv.org/abs/2107.06111 [27]

Funding *Falko Hegerfeld*: Partially supported by DFG Emmy Noether-grant (KR 4286/1).

1 Introduction

The goal of parameterized complexity is to leverage input structure to obtain faster algorithms than in the worst case and to identify algorithmically useful structure. The most prominent structural graph parameter *treewidth* measures the size of *separators* decomposing the graph. Many problems admit fast algorithms relative to treewidth and we can often certify their optimality assuming the *Strong Exponential-Time Hypothesis* (SETH) [7, 11, 12, 13, 43].

17th International Symposium on Parameterized and Exact Computation (IPEC 2022).
Editors: Holger Dell and Jesper Nederlof; Article No. 17; pp. 17:1–17:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Such (conditional) *optimality results* allow us to conduct a precise study of the impact of structure on the running time, whereas otherwise the currently best running time might be an artifact due to the momentary lack of algorithmic tools and not inherent to the structure.

The structure captured by treewidth can be varied in several ways: In the *sparse setting*, we may restrict the interplay of separators and/or allow additional connected components from some graph class $\mathcal{H}$; this yields notions such as treedepth as well as deletion resp. elimination distance to $\mathcal{H}$. In the *dense setting*, we may allow large and dense but structured separators; this yields e.g. clique-width and rank-width. Conceptually, the difference between parameters may be quite large: if the complexity of a problem changes between two parameters, then it is difficult to pinpoint which structural feature has lead to the change in complexity.

We seek to delineate more *exact structural thresholds* between these parameters. This can be done by designing algorithms relative to more permissible parameters or by establishing the same lower bounds relative to more restrictive parameters. We focus on the latter approach in a fine-grained setting, i.e., all considered problems can be solved in time $\mathcal{O}^*(c^k)$[1] for some constant c and parameter k and we determine the precise value of the base c.

For parameters other than treewidth far fewer optimality results are known. In particular, to the best of our knowledge, the only known fine-grained optimality results for NP-hard problems relative to a deletion distance are for r-Coloring [35, 43], its generalization List Homomorphism [52], and isolated results on Vertex Cover [33] and Connected Vertex Cover [9]. The crux is that other lower bound proofs deal with more complex problems (e.g., deletion of vertices, packing of subgraphs, etc.) by copying the same (type of) partial solution over many *noncrossing* separators; this addresses several obstacles but makes the approach unsuitable for deletion distance parameters (or even for treedepth). We show that a much broader range of problems may admit such improved lower bounds by giving the new tight lower bounds for *vertex deletion problems* such as Vertex Cover and Odd Cycle Transversal relative to deletion distance parameters, in both sparse and dense settings.

Sparse Setting. Our main problem of study is Deletion to r-Colorable, i.e., delete as few vertices as possible so that an r-colorable graph remains, which specializes to Vertex Cover for $r = 1$ and to Odd Cycle Transversal for $r = 2$. The first parameterization which we study is the size $|X|$ of a *modulator* $X \subseteq V(G)$, or deletion distance, to treewidth r, i.e., $\mathrm{tw}(G - X) \leq r$. Our main result in the sparse setting is the following.

► **Theorem 1.1.** *If there are $r \geq 2$, $\varepsilon > 0$ such that Deletion to r-Colorable can be solved in time $\mathcal{O}^*((r+1-\varepsilon)^{|X|})$, where X is a modulator to treewidth r, then SETH is false.*[2]

The general construction for Deletion to r-Colorable, $r \geq 2$, does not work for the case $r = 1$, i.e., Vertex Cover, and we fill this gap by providing a simple ad-hoc construction for Vertex Cover parameterized by a modulator to pathwidth 2.

► **Theorem 1.2.** *If there is an $\varepsilon > 0$ such that Vertex Cover can be solved in time $\mathcal{O}^*((2-\varepsilon)^{|X|})$, where X is a modulator to pathwidth 2, then SETH is false.*

These results improve the known lower bounds for Vertex Cover and Odd Cycle Transversal parameterized by pathwidth and provide new tight lower bounds for $r \geq 3$ as a matching upper bound follows from generalizing the known algorithm for Odd Cycle Transversal parameterized by treewidth. Note that in Theorem 1.1 the treewidth bound r

[1] The $\mathcal{O}^*$-notation suppresses factors that are polynomial in the input size.
[2] We assume that an appropriate decomposition is given, thus strengthening the lower bounds.

is the same as the bound r on the number of colors. This treewidth bound, at least for $r = 2$, and the pathwidth bound in Theorem 1.2 cannot be improved due to upper bounds obtained by Lokshtanov et al. [44] for Vertex Cover and Odd Cycle Transversal parameterized by an odd cycle transversal or a feedback vertex set. Lokshtanov et al. [43] asked if the complexity of problems, other than r-Coloring (where a modulator to a single path is already sufficient [35]), relative to treewidth could already be explained with parameterization by feedback vertex set. As argued, this cannot be true for Vertex Cover and Odd Cycle Transversal, so our results are essentially the next best explanation.

Furthermore, the previous two theorems also imply the same lower bound for parameterization by *treedepth*[3], thus yielding the first tight lower bounds relative to treedepth for vertex selection problems and partially resolving a question of Jaffke and Jansen [35] regarding the complexity relative to treedepth for problems studied by Lokshtanov et al. [43].

▶ **Corollary 1.3.** *If there is an $r \geq 1$ and an $\varepsilon > 0$ such that Deletion to r-Colorable can be solved in time $\mathcal{O}^*((r + 1 - \varepsilon)^{\mathrm{td}(G)})$, then SETH is false.*

Dense Setting. Our results on deletion distances can actually be lifted to the *dense setting*. We do so by considering *twinclasses*, which are arguably the simplest form of dense structure. A twinclass is an equivalence class of the *twin*-relation, which says that two vertices u and v are twins if $N(u) \setminus \{v\} = N(v) \setminus \{u\}$, i.e., u and v have the same neighborhood outside of $\{u, v\}$. Given two distinct twinclasses, either all edges between them exist or none of them do. Contracting each twinclass yields the *quotient graph* G^q and we obtain *twinclass-variants* of the usual graph parameters treedepth, cutwidth, pathwidth, and treewidth by measuring these parameters on the quotient graph G^q, e.g., the *twinclass-pathwidth* of G is $\text{tc-pw}(G) = \mathrm{pw}(G^q)$. The parameters twinclass-pathwidth and twinclass-treewidth have been studied before under the name *modular pathwidth* and *modular treewidth* [42, 47, 51]. Furthermore, we remark that the previously studied parameter *neighborhood diversity* satisfies $\mathrm{nd}(G) = |V(G^q)|$ [41]. Relationships between parameters transfer to their twinclass-variants and twinclass-pathwidth is more restrictive than *linear-clique-width*. Similarly, we obtain *twinclass-modulators*, but we measure the complexity of the remaining components on the level of the original graph, i.e., a twinclass-modulator (TCM) $\mathcal{X}$ to treewidth r is a family $\mathcal{X}$ of twinclasses such that $\mathrm{tw}(G - \bigcup_{X \in \mathcal{X}} X) \leq r$. We can now state our second main result, which, similarly to the sparse setting, also carries over to twinclass-treedepth.

▶ **Theorem 1.4.** *If there are $r \geq 2$, $\varepsilon > 0$ such that Deletion to r-Colorable can be solved in time $\mathcal{O}^*((2^r - \varepsilon)^{|\mathcal{X}|})$, where $\mathcal{X}$ is a TCM to treewidth r, then SETH is false.*

Additionally, it follows that if there are $r \geq 2$, $\varepsilon > 0$ such that Deletion to r-Colorable can be solved in time $\mathcal{O}^*((2^r - \varepsilon)^{\text{tc-td}(G)})$, then SETH is false.

Due to the inequalities $\mathrm{cw}(G) \leq \text{tc-pw}(G) + 3$ and $\mathrm{pw}(G) \leq \mathrm{td}(G)$, cf. Lampis [42] and Nešetřil and Ossona de Mendez [48], we see that $\mathrm{cw}(G) \leq \text{tc-td}(G) + 3$. Hence any $\mathcal{O}^*(c^{\mathrm{cw}(G)})$-time algorithm also implies a $\mathcal{O}^*(c^{\text{tc-td}(G)})$-time algorithm. Thus, the following result, relying on standard techniques for dynamic programming on graph decompositions such as the $(\min, +)$-*cover product*, yields a tight upper bound complementing the previous lower bounds.

[3] If $\mathrm{tw}(G - X) \leq t$, then $\mathrm{td}(G) \leq |X| + (t + 1) \log_2 |V|$, cf. Nešetřil and Ossona de Mendez [48], and $\mathcal{O}^*(c^{\mathrm{td}(G)}) = \mathcal{O}^*(c^{|X|} |V|^{(t+1) \log_2 c}) = \mathcal{O}^*(c^{|X|})$.

▶ **Theorem 1.5.** *Given a k-clique-expression μ for G, Deletion to r-Colorable on G can be solved in time $\mathcal{O}^*((2^r)^k)$.*[4]

There is no further lower bound result for Vertex Cover, since $r + 1 = 2^r$ for $r = 1$ and hence Theorem 1.2 already yields a tight lower bound for the clique-width-parameterization.

Going into more detail, the twinclasses of the modulator in the construction for Theorem 1.4 are *true twinclasses*, i.e., each twinclass induces a clique, and moreover they are of size r (with a small exception). Intuitively, allowing for deletions, there are 2^r possible sets of at most r colors that can be assigned to a clique of size r, e.g., the empty set $\emptyset$ corresponds to deleting the clique completely. Hence, our results essentially show that it is necessary and optimal to go through all of these color sets for each twinclass in the modulator.

In contrast, consider the situation for r-Coloring where Lampis [42] has obtained tight running times of $\mathcal{O}^*\left(\binom{r}{\lfloor r/2 \rfloor}^{\text{tc-tw}(G)}\right)$ when parameterized by twinclass-treewidth and of time $\mathcal{O}^*((2^r - 2)^{\text{cw}(G)})$ when parameterized by clique-width. Whereas the complexities for r-Coloring vary between the twinclass-setting and clique-width, this is not the case for Deletion to r-Colorable. The base $\binom{r}{\lfloor r/2 \rfloor}$ is due to the fact that without deletions only color sets of the same size as the considered (true) twinclass can be attained and the most sets are possible when the size is $\lfloor r/2 \rfloor$. For clique-width, a *label class* may induce more complicated graphs than cliques or independent sets and the interaction between two label classes may also be more intricate. Lampis [42] shows that the extremal cases of color sets $\emptyset$ and $[r] = \{1, \dots, r\}$ can be handled separately, thus yielding the base $2^r - 2$ for clique-width.

Additional results. As separate results, we obtain the following four results:

▶ **Theorem 1.6.** *Assuming the SETH, the following lower bounds hold:*

- *Dominating Set cannot be solved in time $\mathcal{O}^*((4 - \varepsilon)^{\text{tc-ctw}(G)})$ for any $\varepsilon > 0$.*
- *Total Dominating Set cannot be solved in time $\mathcal{O}^*((4 - \varepsilon)^{\text{ctw}(G)})$ for any $\varepsilon > 0$.*
- *Maximum Cut cannot be solved in time $\mathcal{O}^*((2 - \varepsilon)^{|X|})$ for any $\varepsilon > 0$, where X is a modulator to treewidth at most 2.*
- *K_r-free Deletion cannot be solved in time $\mathcal{O}^*((2 - \varepsilon)^{|X|})$ for any $\varepsilon > 0$ and $r \geq 3$, where X is a modulator to treewidth at most $r - 1$.*

The first result improves the parameterization of the tight lower bound for Dominating Set obtained by Katsikarelis et al. [38] from linear-clique-width to twinclass-cutwidth. We prove this by reducing Total Dominating Set parameterized by cutwidth to Dominating Set parameterized by twinclass-cutwidth and providing a lower bound construction for Total Dominating Set parameterized by cutwidth.

Lastly, the lower bound for Vertex Cover, Theorem 1.2, also implies tight lower bounds for Maximum Cut and K_r-free Deletion, which again imply the same lower bounds parameterized by treedepth. The former also partially answers a question of Jaffke and Jansen [35], by being another problem considered by Lokshtanov et al. [43] whose running time cannot be improved when parameterizing by treedepth instead of treewidth.

[4] Jacob et al. [34] have simultaneously proven this upper and lower bound for the special case of Odd Cycle Transversal, $r = 2$, parameterized by clique-width. Their construction also proves the lower bound for linear-clique-width, but not for the more restrictive twinclass-treedepth or twinclass-modulator like our construction.

Technical contribution. We start by recalling the standard approach of Lokshtanov et al. [43] to proving tight lower bounds for problems parameterized by pathwidth at a high level. Given a Satisfiability instance σ, the variables are partitioned into t groups of constant size. For each variable group, a *group gadget* is constructed that can encode all assignments of this variable group into partial solutions of the considered target problem. The group gadget usually consists of a bundle of long path-like gadgets inducing a sequence of *disjoint* separators. Further gadgets attached to these separators decode the partial solutions and check whether the corresponding assignment satisfies some clause. Ideally, the path gadgets are designed so that a partial solution transitions through a well-defined sequence of states when viewed at consecutive separators. For most problems, the gadgets do not behave this nicely though. For example, in Odd Cycle Transversal it is locally always preferable to delete a vertex instead of not deleting it. Such behavior leads to undesired state changes called *cheats*, but for appropriate path gadgets there can only be a constant number of cheats on each path. By making the path gadgets long enough, one can then find a region containing no cheats where we can safely decode the partial solutions.

For problems such as r-Coloring, all states are equally constraining and such cheats do not occur, hence enabling us to prove the same lower bounds under more restrictive parameters such as feedback vertex set. But for *vertex deletion problems*, like Odd Cycle Transversal, these cheats do occur and pose a big issue when trying to compress the path gadgets into a single separator X, since deletions in X are highly favorable. On a single separator X such behavior means that one partial solution is *dominating* another and if we cannot control this behavior, then we lose the dominated partial solution for the purpose of encoding group assignments. Concretely, for Odd Cycle Transversal we obtain dominating partial solutions by deleting further vertices in the single separator X. The number of deletions is bounded from above by the *budget constraint*, but if we limit the number of deletions in X, then we do not have $3^{|X|}$ partial solutions anymore and the construction may not be able to attain the desired base in the running time.

To resolve this issue we expand upon a technique of Cygan et al. [9] and construct an instance with a slightly large parameter value, i.e., a slightly larger single separator X. Thus, we can limit the number of deletions and are still able to encode sufficiently many group assignments. More precisely, we consider only partial solutions with the same number of deletions in X, hence only pairwise non-dominating partial solutions remain. We construct a *structure gadget* to enforce a lower bound on the number of deletions in X. A positive side effect is that the remaining gadgets can also leverage the structure of the partial solutions.

In the dense setting and especially for a higher number r of colors, this issue is amplified. Here, we consider the states of twinclasses, instead of single vertices, in a partial solution. For a twinclass, there is a hierarchy of dominating states: any state that does not delete all vertices in the twinclass is dominated by a state that deletes further vertices in the twinclass. For Deletion to r-Colorable, the maximum number of states is achieved on a true twinclass of size r and we can partition the states into *levels* based on the number of deletions they induce. Within each level, the states are pairwise non-dominating. Consequently, we restrict the family of partial solutions so that for every level the number of twinclasses with that level is fixed. This requires a considerably more involved construction of the structure gadget which now has to distinguish states based on their level.

Related work. There is a long line of work relative to treewidth [2, 7, 11, 12, 15, 16, 17, 21, 39, 43, 46, 45, 49, 50] and all of these lower bounds, except for the result by Egri et al. [17], already apply to pathwidth. In the sparse setting, there is further work on the parameterization by *cutwidth* [8, 26, 37, 45, 52, 53] and by feedback vertex set [43, 52]. We remark that the works of van Geffen et al. [53] and Piecyk and Rzążewski [52] show that previous lower bounds relative to pathwidth already hold for more restrictive parameterizations. In the dense setting, there are some results [32, 34, 38, 42] on parameterization by clique-width and these lower bounds already apply to linear-clique-width, but not to the more restrictive parameters that we consider. The work by Iwata and Yoshida [32] also provides equivalences between different lower bounds and works under a weaker assumption than SETH, unfortunately their techniques blow up the modulator too much and are not applicable in our case. Finally, the complexity of r-Coloring and the more general homomorphism problems has been extensively studied [17, 21, 24, 35, 42, 49, 50, 52], only two of these articles [24, 42] consider the dense setting. Jaffke and Jansen [35] closely study the complexity of r-Coloring parameterized by the deletion distance to various graph classes $\mathcal{F}$; in particular, the base for treewidth can already be explained by deletion distance to a single path.

On the algorithmic side, the study of heterogeneous parameterizations has been gaining traction [3, 4, 18, 20, 19, 29, 36], yielding the notions of $\mathcal{H}$-treewidth and $\mathcal{H}$-*elimination distance*, which is a generalization of treedepth. Currently, only few of these works [18, 36] contain algorithmic results that are sufficiently optimized to apply to our fine-grained setting. Jansen et al. [36] show that Vertex Cover can be solved in time $\mathcal{O}^*(2^k)$ and Odd Cycle Transversal in time $\mathcal{O}^*(3^k)$ when parameterized by bipartite-treewidth. Eiben et al. [18] show that Maximum Cut can be solved in time $\mathcal{O}^*(2^k)$ when parameterized by $\mathcal{R}_w$-treewidth, where $\mathcal{R}_w$ denotes the graphs of *rank-width* at most w.

Another line of work is on depth-parameters in the dense setting [5, 14, 22, 23, 25, 28, 40] such as *shrub-depth* and *sc-depth*. The algorithmic results relative to these parameters are largely concerned with meta-results so far [5, 25] and their relation to clique-width is not strong enough to preserve the complexity in our fine-grained setting.

Organization. We discuss the preliminaries and basic notation in Section 2. The relationships between the considered parameters are discussed in Section 3. In Section 4, we give an outline of our two main results: the lower bound for Deletion to r-Colorable in the sparse setting and the dense setting. The algorithm for Deletion to r-Colorable parameterized by clique-width is given in Section 5. We conclude in Section 6. Appendix A contains the formal definitions of the considered problems. The remaining results, including the missing proofs, can be found in the full version of the paper [27]. ·

2 Preliminaries

If n is a positive integer, we define $[n] = \{1, \dots, n\}$. If S is a set, we define $\mathcal{P}(S) = \{T \subseteq S\}$ and if $0 \leq k \leq |S|$, we define $\binom{S}{k} = \{T \subseteq S : |T| = k\}$ and $\binom{S}{\leq k} = \{T \subseteq S : |T| \leq k\}$ and $\binom{S}{\geq k}$ analogously. If $0 \leq k \leq n$, we define $\binom{n}{\leq k} = |\binom{[n]}{\leq k}|$ and similarly $\binom{n}{\geq k}$. If $\mathcal{S}$ is a set family, we define $\bigcup(\mathcal{S}) = \bigcup_{S \in \mathcal{S}} S$. If $f\colon A \to C$ is a function and $B \subseteq A$, then $f\big|_B$ denotes the *restriction* of f to B. If $f, g\colon A \to B$ are two functions, we write $f \equiv g$ if $f(a) = g(a)$ for all $a \in A$. If p is a boolean predicate, we let $[p]$ denote the *Iverson bracket* of p, which is 1 if p is true and 0 if p is false.

We use common graph-theoretic notation and assume that the reader knows the essentials of parameterized complexity. Let $G = (V, E)$ be an undirected graph. For a vertex set $X \subseteq V$, we denote by $G[X]$ the subgraph of G that is induced by X. The *open neighborhood* of a vertex v is given by $N(v) = \{u \in V : \{u, v\} \in E\}$, whereas the *closed neighborhood* is given by

$N[v] = N(v) \cup \{v\}$. For sets $X \subseteq V$ we define $N[X] = \bigcup_{v \in X} N[v]$ and $N(X) = N[X] \setminus X$. For two disjoint vertex subsets $A, B \subseteq V$, adding a *join* between A and B means adding all edges between A and B. For a vertex set $X \subseteq V$, we define $\delta(X) = \{\{x, y\} \in E : x \in X, y \notin X\}$.

An *r-coloring* of a graph $G = (V, E)$ is a function $\varphi \colon V \to [r]$ such that $\varphi(u) \neq \varphi(v)$ for all $\{u, v\} \in E$. We say that G is *r-colorable* if there is an r-coloring of G. The *chromatic number* of G, denoted by $\chi(G)$, is the minimum r such that G is r-colorable.

Quotients and twins. Let Π be a partition of $V(G)$. The *quotient graph* G/Π is given by $V(G/\Pi) = \Pi$ and $E(G/\Pi) = \{\{B_1, B_2\} : \exists u \in B_1, v \in B_2 \colon \{u, v\} \in E(G)\}$. We say that two vertices u, v are *twins* if $N(u) \setminus \{v\} = N(v) \setminus \{u\}$. The equivalence classes of this relation are called *twinclasses*. More specifically, if $N(u) = N(v)$, then u and v are *false twins* and if $N[u] = N[v]$, then u and v are *true twins*. Every twinclass of size at least 2 consists of only false twins or only true twins. A false twinclass induces an independent set and a true twinclass induces a clique. Let $\Pi_{tc}(G)$ be the partition of $V(G)$ into twinclasses.

2.1 Graph Parameters

Sparse Parameters. The definition of treewidth, pathwidth, treedepth, and cutwidth are standard and can be found in the full version. We will construct graphs that have small treewidth except for one central part. This structure is captured by the concept of a *modulator*. We say that $X \subseteq V(G)$ is a *modulator to treewidth/pathwidth r* for G if $\mathrm{tw}(G - X) \leq r$ or $\mathrm{pw}(G - X) \leq r$, respectively.

Lifting to Twinclasses

We define the *twinclass-treewidth*, *twinclass-pathwidth*, *twinclass-treedepth*, and *twinclass-cutwidth* of G by $\text{tc-tw}(G) = \mathrm{tw}(G/\Pi_{tc}(G))$, $\text{tc-pw}(G) = \mathrm{pw}(G/\Pi_{tc}(G))$, $\text{tc-td}(G) = \mathrm{td}(G/\Pi_{tc}(G))$, and $\text{tc-ctw}(G) = \mathrm{ctw}(G/\Pi_{tc}(G))$, respectively. The parameters twinclass-treewidth and twinclass-pathwidth have been considered before under the name modular treewidth and modular pathwidth [42, 47, 51]. We prefer to use the prefix twinclass instead of modular to distinguish from the case where one works with the quotient graph arising from the *modular partition* of G.

▶ **Definition 2.1.** Let $G = (V, E)$ be a graph. A *twinclass-modulator (TCM)* $\mathcal{X} \subseteq \Pi_{tc}(G)$ of G to treewidth r is a set of twinclasses of G such that $\mathrm{tw}(G - \bigcup(\mathcal{X})) \leq r$. The *size* of a twinclass-modulator $\mathcal{X}$ is $|\mathcal{X}|$, i.e., the number of twinclasses $\mathcal{X}$ contains.

Clique-Width

A *labeled graph* is a graph $G = (V, E)$ together with a label function $\ell \colon V \to \mathbb{N} = \{1, 2, 3, \ldots\}$. We say that a labeled graph is *k-labeled* if $\ell(v) \leq k$ for all $v \in V$. For a label i, we denote by G^i the subgraph induced by the vertices with label i, i.e. $G^i = G[\ell^{-1}(i)]$. We consider the following three operations on labeled graphs: the *union*-operation $\mathrm{union}(G_1, G_2)$ constructs the disjoint union of two labeled graphs G_1 and G_2; the *relabel*-operation $\mathrm{lab}_{i \to j}(G)$ changes the label of all vertices in G with label i to label j; the *join*-operation $\mathrm{join}_{i,j}(G)$, $i \neq j$, adds all possible edges between vertices in G with label i and vertices in G with label j. As a base case, we have the *introduce*-operation $\mathrm{in}_i(v)$ which constructs a single-vertex graph whose unique vertex v has label i. A valid expression that only consists of introduce-, union-, relabel-, and join-operations is called a *clique-expression*. The labeled graph constructed by a clique-expression μ is denoted $G(\mu)$. To a clique-expression μ we associate a syntax tree $\mathcal{T}_\mu$

in the natural way and to each node $t \in V(\mathcal{T}_\mu)$ the corresponding operation. For any node $t \in V(\mathcal{T}_\mu)$, the subtree rooted at t induces a subexpression μ_t and we define $G_t = G(\mu_t)$ as the labeled graph constructed by μ_t.

We say that a clique-expression σ is a *k-clique-expression* or just *k-expression* if G_t is k-labeled for all $t \in V(\mathcal{T}_\mu)$. The *clique-width* of a graph G, denoted by $\mathrm{cw}(G)$, is the minimum k such that there exists a k-expression μ such that G is isomorphic to $G(\mu)$ after forgetting the labels. A clique-expression μ is *linear* if in every union-operation the second graph consists only of a single vertex. Accordingly, we also define the *linear-clique-width* of a graph G, denoted $\text{lin-cw}(G)$, by only considering linear clique-expressions.

2.2 Strong Exponential-Time Hypothesis

For our lower bounds, we assume the *Strong Exponential-Time Hypothesis* (SETH) [31] which concerns the complexity of q-Satisfiability, i.e., Satisfiability where all clauses contain at most q literals. Let $c_q = \inf\{\delta : q\text{-Satisfiability can be solved in time } \mathcal{O}(2^{\delta n})\}$ for all $q \geq 3$. The weaker *Exponential-Time Hypothesis* (ETH) of Impagliazzo and Paturi [30] posits that $c_3 > 0$, whereas the Strong Exponential-Time Hypothesis states that $\lim_{q \to \infty} c_q = 1$. When proving lower bounds based on SETH, we make use of the following equivalent formulations.

▶ **Theorem 2.2** ([9]). *The following statements are equivalent to SETH:*

1. *For all $\delta < 1$, there is a clause size q such that q-Satisfiability cannot be solved in time $\mathcal{O}(2^{\delta n})$, where n is the number of variables.*
2. *For all $\delta < 1$, there is a set size q such that q-Hitting Set, i.e., all sets contain at most q elements, cannot be solved in time $\mathcal{O}(2^{\delta n})$, where n is the universe size.*

3 Relations between Parameters

In this section we discuss the relationships between the parameters considered in this article.

▶ **Lemma 3.1** ([1], Chapter 6 of [48]). *For any graph G, we have that $\mathrm{tw}(G) \leq \mathrm{pw}(G) \leq \mathrm{td}(G) - 1$, $\mathrm{td}(G) \leq (\mathrm{tw}(G) + 1)\log_2 |V(G)|$, $\mathrm{tw}(G) \leq \mathrm{pw}(G) \leq \mathrm{ctw}(G)$, and $\mathrm{td}(G) \leq \mathrm{td}(G - v) + 1$ for any vertex $v \in V(G)$. These inequalities come with algorithms that can transform the appropriate decomposition in polynomial time.*

▶ **Corollary 3.2.** *For any graph $G = (V, E)$ and $c, r \in \mathbb{N}$, if there is a modulator $X \subseteq V$ to treewidth r, i.e., $\mathrm{tw}(G - X) \leq r$, then we have that $\mathrm{td}(G) \leq |X| + (r + 1)\log_2 |V|$. In particular, we have that $\mathcal{O}^*(c^{\mathrm{td}(G)}) \leq \mathcal{O}^*(c^{|X|})$ for all $c \geq 1$. The decompositions can be transformed in polynomial time.*

Proof. Let X be a modulator to treewidth r for G. By Lemma 3.1, we see that $\mathrm{td}(G - X) \leq (r + 1)\log_2 |V|$ for $G - X$. By repeatedly invoking the inequality $\mathrm{td}(G) \leq \mathrm{td}(G - v) + 1$ for $v \in X$, we obtain $\mathrm{td}(G) \leq |X| + (r + 1)\log_2 |V|$. To see the claim regarding the $\mathcal{O}^*$-notation, we compute $\mathcal{O}^*(c^{\mathrm{td}(G)}) = \mathcal{O}^*(c^{|X|}|V|^{(r+1)\log_2 c}) = \mathcal{O}^*(c^{|X|})$. ◀

▶ **Theorem 3.3.** *Let $G = (V, E)$ be a graph. We have the following two chains of inequalities:*

$$\mathrm{cw}(G) \leq \text{lin-cw}(G) \leq \text{tc-pw}(G) + 3 \leq \text{tc-td}(G) + 2 \leq \mathrm{td}(G) + 2,$$
$$\mathrm{cw}(G) \leq \text{lin-cw}(G) \leq \text{tc-pw}(G) + 3 \leq \text{tc-ctw}(G) + 3 \leq \mathrm{ctw}(G) + 3.$$

Proof. Follows from [42, Lemma 2.1], Lemma 3.1 and the last inequalities in both rows follow from the fact that $G/\Pi_{tc}(G)$ is a subgraph of G and that treedepth and cutwidth are subgraph-monotone. ◀

▶ **Lemma 3.4.** *Suppose that G admits a TCM $\mathcal{X}$ to treewidth r, then* $\text{tc-td}(G) \leq |\mathcal{X}| + (r+1)\log_2 |V(G/\Pi_{tc}(G))|$. *In particular, we have for any $c \geq 1$ that $\mathcal{O}^*(c^{\text{tc-td}(G)}) \leq \mathcal{O}^*(c^{|\mathcal{X}|})$. The decompositions can be transformed in polynomial time.*

Proof. Since $G/\Pi_{tc}(G) - \mathcal{X}$ is an induced subgraph of $G - \bigcup(\mathcal{X})$, we see that $\text{tw}(G/\Pi_{tc}(G) - \mathcal{X}) \leq \text{tw}(G - \bigcup(\mathcal{X})) \leq r$. The remainder of the proof is analogous to Corollary 3.2 by working on the quotient graph $G/\Pi_{tc}(G)$. ◀

4 Outline of Main Result

We outline our two main results, i.e., tight lower bounds for Deletion to r-Colorable parameterized by a (twinclass-)modulator to treewidth r. Conceptually, the constructions for the sparse setting and for the dense setting are similar. The most significant change is in the *structure gadget*, since we have to enforce a considerably more involved structure in the dense setting. We give an overview of both settings and go into more detail for the dense case.

We fix the number of colors $r \geq 2$. *Solutions* are functions $\varphi\colon V(G) \to [r] \cup \{\bot\}$ so that for every edge $\{u, v\} \in E(G)$ either $\varphi(u) = \varphi(v) = \bot$ or $\varphi(u) \neq \varphi(v)$. Hence, $\varphi^{-1}(\bot)$ is the set of deleted vertices, whereas $\varphi|_{V(G)\setminus\varphi^{-1}(\bot)}$ is an r-coloring of the remaining graph.

In both settings we want to simulate a *logical OR* constraint. For Odd Cycle Transversal, i.e. $r = 2$, we can use *odd cycles*. For $r \geq 3$, Theorem 4.1 provides an analogue, where a graph H is $(r+1)$-*critical* if $\chi(H) = r + 1$ and $\chi(H - v) = r$ for all $v \in V(H)$.

▶ **Theorem 4.1** (proof in full version)**.** *There exists a family $\mathcal{H}^r$ of $(r+1)$-critical graphs with treewidth r such that for every $s \in \mathbb{N}$, there exists a graph $H \in \mathcal{H}^r$ with $s \leq |V(H)| \leq s + r$.*

Setup. Given a q-Satisfiability instance σ with n variables and m clauses, we start with the following standard step [43]: we partition the variables into $t = \lceil n/p_0 \rceil$ groups of size p_0, where p_0 only depends on the running time base that we want to rule out. Furthermore, we pick an integer p depending on p_0 that represents the size of the groups in the graph.

4.1 Sparse Setting

Central vertices and solution structure. We construct a graph G that has a solution φ for Deletion to r-Colorable with cost $|\varphi^{-1}(\bot)| \leq b$ if and only if σ is satisfiable. Converting from base $r+1$ to base 2 implies that G should admit a modulator X to treewidth r of size roughly $n \log_{r+1}(2)$. Like Cygan et al. [9], we make the modulator slightly larger, thus picking a larger p. The modulator X consists of $t + 1$ vertex groups: the first t groups U_i, $i \in [t]$, are independent sets of size p each and correspond to the variable groups; the last group F is a clique of size r which simulates List Coloring constraints.

On each group U_i, we consider the set of partial solutions $\Phi_i = \{\varphi\colon U_i \to [r] \cup \{\bot\} : |\varphi^{-1}(\bot)| = p/(r+1)\}$. By picking p large enough, Φ_i is sufficiently large to encode all assignments of the i-th variable group. Defining Φ_i in this way achieves two things: first, the solutions in Φ_i are pairwise non-dominating; secondly, this fixes the budget used on the modulator. The second point is important, because by also fixing the budget on the remaining graph via a vertex-disjoint packing $\mathcal{P}$ of $(r+1)$-critical graphs, no vertex of F can be deleted, which allows us to simulate List Coloring constraints with the clique F.

Structure gadgets. The next step is to enforce that only the solutions in Φ_i can be attained on group U_i. By choosing the budget b appropriately, we obtain an upper bound on the number of deletions in U_i. To obtain a lower bound, we construct the *structure gadgets*. These are built by combining $(r+1)$-critical graphs with the *arrow* gadget of Lokshtanov et al. [43]. A (thin) arrow simply propagates a deletion from a vertex u to another vertex v; else if u is not deleted, then v is not deleted and the arrow does not affect the remaining graph.

The structure gadget works as follows: if φ deletes less than $p/(r+1)$ vertices in group U_i, then there is a subset $S \subseteq U_i$ of size $|S| = (|U_i| - p/(r+1)) + 1$ that avoids all deletions in U_i. For every subset of this size, G contains a $(r+1)$-critical graph $L_{i,S}$ with an arrow from every $u \in S$ to a private vertex v in $L_{i,S}$, hence simulating an OR on the vertices in S. Since S avoids all deletions of φ, no deletion is propagated to $L_{i,S}$ and φ must pay extra to resolve $L_{i,S}$. By copying each $L_{i,S}$ sufficiently often, we can ensure that the existence of a deletion-avoiding S implies that φ must exceed our budget constraint.

Decode and verify. The remaining construction decodes the partial solution on the modulator X and verifies if the corresponding truth assignment satisfies all clauses of σ. One could generalize the gadgets of Lokshtanov et al. [43] to higher r, but this leads to an involved construction with a worse bound on the treewidth of the remainder: for Odd Cycle Transversal the construction of Lokshtanov et al. has treewidth 4, whereas the simpler construction we use has only treewidth 2. More details will be presented in the dense case.

4.2 Dense Setting

We now have a twinclass-modulator $\mathcal{X}$ to treewidth r instead of a basic modulator and this changes the possible states as follows. Whereas φ could assume $r+1$ different states on a single vertex u, i.e., one of the r colors or deleting the vertex, there are 2^r possible states on a true twinclass U of size r; each corresponds to a possible value of $\varphi(U) \setminus \{\bot\} \subseteq [r]$. Since U is a true twinclass, no color is used multiple times and the exact mapping $\varphi\big|_U$ is irrelevant.

Central twinclasses and setup. The twinclass-modulator $\mathcal{X}$ of the constructed graph G consists of $t+1$ groups and each group is a family of twinclasses. The first t groups $\mathcal{U}_i$, $i \in [t]$, correspond to the variable groups and each consists of p true twinclasses of size r that are pairwise non-adjacent. The last group contains the clique F.

Solution structure. Our family Φ_i of considered partial solutions on group $\mathcal{U}_i$ should achieve the same two things as before. First, consider the structure of states of φ on a twinclass $U \in \mathcal{U}_i$ precisely: fix a state $C = \varphi(U) \setminus \{\bot\}$ and note that all states $C' \subsetneq C$ *dominate* C if we disregard the budget constraint, i.e., φ remains a solution if we replace C by C'. After arranging the states into *levels* according to the number ℓ of deleted vertices, there is no domination between states on the same level. This motivates the following definition.

▶ **Definition 4.2** (informal)**.** Given rationals $0 < c_\ell < 1$, $\ell \in \{0\} \cup [r]$, with $\sum_{\ell=0}^{r} c_\ell = 1$, the set Φ_i consists of solutions φ on the family of twinclasses $\mathcal{U}_i$ such that for every $\ell \in \{0\} \cup [r]$ there are exactly $c_\ell \cdot |\mathcal{U}_i|$ twinclasses $U \in \mathcal{U}_i$ where φ deletes exactly ℓ vertices in U.

Essentially, we are only restricting how the deletions can be distributed inside the modulator; there are no restrictions on the used colors. This again fixes the budget used on the modulator, allowing us to simulate List Coloring constraints with the clique F. By picking $c_\ell = \binom{r}{\ell}2^{-r}$, $\ell \in \{0\} \cup [r]$, we ensure that Φ_i contains the solutions on $\mathcal{U}_i$ where all 2^r states appear the same number of times. This enables us to choose p small enough so that the time calculations work out and simultaneously large enough so that an injective mapping $\kappa_i\colon \{0,1\}^{p_0} \to \Phi_i$, mapping truth assignments of the i-th variable group to solutions in Φ_i, exists.

Thick arrows and structure gadgets. To enforce the structure of Φ_i, we need a gadget to distinguish different number of deletions inside a twinclass. We can construct such a gadget $A_\ell(U,v)$, $\ell \in [r]$, also called *thick ℓ-arrow*. See Lemma 4.3 for the gadget's behavior.

▶ **Lemma 4.3** (informal). *Let U be a set of r true twins and v be a vertex that is not adjacent to U and $\ell \in [r]$. There is a gadget $A = A_\ell(U,v)$ of treewidth r with the following properties:*

- *Any solution φ must delete at least ℓ vertices in $A - U$.*
- *If a solution φ deletes exactly ℓ vertices in $A - U$, then φ can only delete v if φ deletes at least ℓ vertices in U.*

We proceed by constructing the *structure gadgets* which enforce that the partial solution on $\mathcal{U}_i$ belongs to Φ_i. Let $c_{<\ell} = c_0 + \cdots + c_{\ell-1}$ for all $\ell \in \{0\} \cup [r]$. For every group $i \in [t]$, number of deletions $\ell \in [r]$, set of twinclasses $\mathcal{S} \subseteq \mathcal{U}_i$ with $|\mathcal{S}| = c_{<\ell} \cdot p + 1$, we add an $(r+1)$-critical graph $L_{i,\ell,\mathcal{S}} \in \mathcal{H}^r$ consisting of at least $|\mathcal{S}|$ vertices. For every $U \in \mathcal{S}$, we pick a private vertex v in $L_{i,\ell,\mathcal{S}}$ and add the thick ℓ-arrow $A_\ell(U,v)$. We create a large number of copies of each $L_{i,\ell,\mathcal{S}}$ and the incident thick arrows.

The number of deletions in the central vertices is already bounded from above by the budget constraint. If too few deletions occur in the twinclasses of $\mathcal{U}_i$, then we can find an ℓ and an $\mathcal{S} \subseteq \mathcal{U}_i$ with $|\mathcal{S}| = c_{<\ell} \cdot p + 1$ such that less than ℓ vertices are deleted in each $U \in \mathcal{S}$. Hence, all thick ℓ-arrows leading to $L_{i,\ell,\mathcal{S}}$ and its copies cannot propagate deletions. To resolve all these $(r+1)$-critical graphs, one extra vertex per copy must be deleted. Due to the large number of copies, this implies that we must violate our budget constraint.

Hence, for any $\mathcal{S} \subseteq \mathcal{U}_i$ with $|\mathcal{S}| = c_{<\ell} \cdot p + 1$ and any solution φ obeying the budget constraint there is at least one twinclass $U \in \mathcal{S}$ in which φ deletes at least ℓ vertices. Therefore, there are at least $(1 - c_{<\ell})p$ twinclasses in $\mathcal{U}_i$ where φ deletes at least ℓ vertices. Since this holds for all $\ell \in \{0\} \cup [r]$ and the budget b is chosen appropriately, all inequalities have to be tight and the deletions inside $\mathcal{U}_i$ follow the distribution imposed by Φ_i.

Color-set-gadgets and decoding gadgets. Next, we discuss the decoding part of the construction. Since gadgets cannot read the color of single vertices but only of a whole twinclass, we need *color-set-gadgets* to detect the colors used on a twinclass, cf. Lemma 4.4.

▶ **Lemma 4.4** (informal). *Let U be a set consisting of r true twins and v be a vertex that is not adjacent to U and let $C \subsetneq [r]$. There is a gadget $B = B_C(U,v)$ of treewidth r such that:*

- *Any solution φ deletes at least $(r - |C|) + 1$ vertices in $B - U$.*
- *If φ deletes exactly $(r - |C|) + 1$ vertices in $B - U$, then $\varphi(v) = \bot$ only if $\varphi(U) \setminus \{\bot\} \subseteq C$.*

To construct the color-set-gadgets we rely on the List Coloring constraints that are simulated with the central clique F. Note that the color-set-gadgets only check for set inclusion and not set equality. Using the structure of solutions in Φ_i however, the color-set-gadgets will still be sufficient to distinguish the solutions in Φ_i from each other.

By using a complete $(r+1)$-partite graph with all sets of the partition being singletons except for one large independent set, we can simulate a logical AND, see Lemma 4.5.

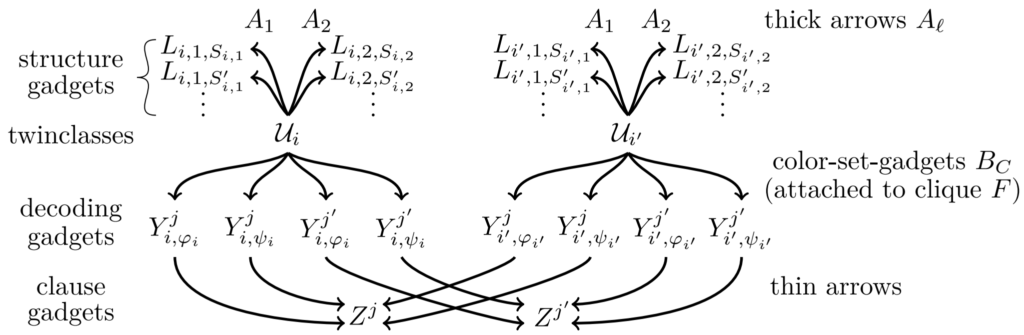

Figure 1 An overview of the construction for the dense setting in case of $r = 2$. The arrows point in the direction that deletions are propagated by the corresponding gadget.

▶ **Lemma 4.5** (informal). *Let n_Y be a positive integer. There is a gadget Y of treewidth r with a set of input vertices $V' \subseteq V(Y)$, $|V'| = n_Y$, and a vertex $\hat{y} \in V(Y) \setminus V'$ such that:*

- *Any solution φ has to delete at least one vertex in $Y - V'$.*
- *If φ deletes exactly one vertex in $Y - V'$, then $\varphi(\hat{y}) = \bot$ only if $\varphi(V') = \{\bot\}$.*

For the j-th clause, variable group $i \in [t]$, solution $\varphi_i \in \Phi_i$, we invoke Lemma 4.5 to create a gadget Y^j_{i,φ_i} for $n_Y = (1 - c_r)p = (1 - 2^{-r})p$ input vertices and with distinguished vertex $\hat{y}^j_{i,\varphi_i}$. For every twinclass $U \in \mathcal{U}_i$ with $\varphi_i(U) \neq [r]$, we pick a private input vertex v of Y^j_{i,φ_i} and add the color-set-gadget $B_{\varphi_i(U)\setminus\{\bot\}}(U, v)$. By Lemma 4.5, the vertex $\hat{y}^j_{i,\varphi_i}$ can only be deleted if all input vertices of Y^j_{i,φ_i} are deleted. Due to Lemma 4.4 and the structure of Φ_i, this will only be the case if φ_i is the partial solution on $\mathcal{U}_i$.

Clause gadgets. For the j-th clause, we add an $(r + 1)$-critical graph $Z^j \in \mathcal{H}^r$ consisting of at least $q2^{p_0}$ vertices. For every group $i \in [t]$ and solution $\varphi_i \in \Phi_i$ such that $\kappa_i^{-1}(\varphi_i)$ is a partial truth assignment satisfying the j-th clause, we pick a private vertex v in Z^j and add a thin arrow from $\hat{y}^j_{i,\varphi_i}$ to v. The budget constraint will ensure that the only way to delete a vertex in Z^j is by propagating a deletion via a thin arrow from some $\hat{y}^j_{i,\varphi_i}$. By construction of the decoding and clause gadgets this is only possible if the partial solution on $\mathcal{U}_i$ corresponds to a satisfying assignment of the j-th clause. This concludes the construction, cf. Figure 1.

Budget and packing. The budget $b = b_0 + \mathrm{cost}_{\mathcal{P}}$ of the constructed instance (G, b) consists of two parts; $b_0 = trp/2$ is allocated to the central twinclasses and matches the number of deletions incurred by picking a partial solution $\varphi_i \in \Phi_i$ on $\mathcal{U}_i$ for each group $i \in [t]$; the second part $\mathrm{cost}_{\mathcal{P}}$ is due to a vertex-disjoint packing $\mathcal{P}$ which we describe next. A part of each thin arrow in G is added to $\mathcal{P}$ and for every thick arrow, color-set-gadget, or decoding gadget, we add the appropriate parts to $\mathcal{P}$ given by Lemmas 4.3, 4.4, 4.5, respectively. Summing up the implied costs yields $\mathrm{cost}_{\mathcal{P}}$. Hence, we know how the deletions are distributed throughout the various gadgets. In particular, this ensures that no vertex of the central clique F is deleted.

Theorem 1.4 follows by using these ideas and working out the remaining technical details.

5 Algorithm for Deletion to r-Colorable

In this section we describe how to solve Deletion to r-Colorable in time $\mathcal{O}^*((2^r)^k)$ if we are given a k-expression μ for G. We perform bottom-up dynamic programming along the syntax tree $\mathcal{T}_\mu$. We again view *solutions* to Deletion to r-Colorable as functions $\varphi\colon V(G) \to [r] \cup \{\bot\}$ with the property discussed in the outline, cf. Section 4.

▶ **Theorem 5.1.** *Given a k-expression μ for G,* Deletion to r-Colorable *on G can be solved in time $\mathcal{O}^*((2^r)^k)$.*

Proof. Let (G, b) be a Deletion to r-Colorable instance and μ a k-expression for G. We can without of loss of generality assume that μ consists of $\mathcal{O}(|V(G)|)$ union-operations and $\mathcal{O}(|V(G)|k^2)$ unary operations [6]. For every node $t \in V(\mathcal{T}_\mu)$ and label i, we store the set of colors used on G_t^i. After deleting the appropriate vertices, the remaining graph should be r-colorable, hence the possible color sets are precisely the subsets of $[r]$, where $\emptyset$ indicates that all vertices are deleted. Since we use at most k labels at every node, this yields $(2^r)^k$ possible types of partial solutions at each node. If the work for each type is only polynomial, then the claimed running time immediately follows, since there are only a polynomial number of nodes in $V(\mathcal{T}_\mu)$.

For every $t \in V(\mathcal{T}_\mu)$ and $f \colon [k] \to \mathcal{P}([r])$, we consider the set of partial solutions

$$\mathcal{Q}_t[f] = \{\varphi \colon V(G_t) \to [r] \cup \{\bot\} : \varphi \text{ induces an } r\text{-coloring of } G_t - \varphi^{-1}(\bot) \text{ and } \varphi(V(G_t^i)) \setminus \{\bot\} = f(i) \text{ for all } i \in [k]\}$$

and we want to compute the quantity $A_t[f] = \min\{|\varphi^{-1}(\bot)| : \varphi \in \mathcal{Q}_t[f]\}$. Let t_0 be the root node of the k-expression μ. We answer yes if there is an f such that $A_{t_0}[f] \leq b$; otherwise we answer no.

Note that $f(i) = \emptyset$ implies $\varphi(V(G_t^i)) = \{\bot\}$ for all $\varphi \in \mathcal{Q}_t[f]$, i.e., all vertices with label i are deleted. Furthermore, the definition of $\mathcal{Q}_t[f]$ implies that $\mathcal{Q}_t[f] = \emptyset$ and $A_t[f] = \infty$ whenever $|f(i)| > |V(G_t^i)|$ for some $i \in [k]$, we will not explicitly mention this edge case again in what follows and assume that the considered f satisfy $|f(i)| \leq |V(G_t^i)|$ for all $i \in [k]$. We proceed by presenting the recurrences to compute $A_t[f]$ for all t and f and afterwards show the correctness of these recurrences.

Base case. If $t = \text{in}_i(v)$ for some $i \in [k]$, then $A_t[f] = [f(i) = \emptyset]$, because the solution cost is 1 if v is deleted and 0 otherwise.

Relabel case. If $t = \text{lab}_{i \to j}(G_{t'})$ for some $i \neq j \in [k]$ and where t' is the child of t, then

$$A_t[f] = \min\{A_{t'}[f'] : f'(a) = f(a) \text{ for all } a \in [k] \setminus \{i, j\} \text{ and } f'(i) \cup f'(j) = f(j)\}.$$

By assumption, f' will always satisfy $f'(i) = \emptyset$ here, since there are no vertices with label i in $G_{t'}$. This recurrence goes over all ways how the colors $f'(j)$ used for vertices with label j in $G_{t'}$ can be split among the vertices with label i and j in the previous graph G_t. Observe that we are taking the minimum over at most $(2^r)^2 = \mathcal{O}(1)$ numbers on the right-hand side, hence this recurrence can be computed in polynomial time.

Join case. If $t = \text{join}_{i,j}(G_{t'})$ for some $i \neq j \in [k]$, where t' is the child of t, and assuming without loss of generality that $V(G_{t'}^i) \neq \emptyset$ and $V(G_{t'}^j) \neq \emptyset$, then

$$A_t[f] = \begin{cases} A_{t'}[f] & \text{if } f(i) \cap f(j) = \emptyset, \\ \infty & \text{else.} \end{cases}$$

This recurrence filters out all partial solutions where the coloring properties are not satisfied at some newly added edge. This happens precisely when $f(i) \cap f(j) \neq \emptyset$, because then there exists an edge in the join between label i and j whose endpoints get the same color.

Union case. If $t = \text{union}(G_{t_1}, G_{t_2})$ where t_1 and t_2 are the children of t, then

$$A_t[f] = \min\{A_{t_1}[f_1] + A_{t_2}[f_2] : f_1(a) \cup f_2(a) = f(a) \text{ for all } a \in [k]\}.$$

Here, we assume that $\infty + x = x + \infty = \infty + \infty = \infty$ for all $x \in \mathbb{N}$. This recurrence goes for each label $a \in [k]$ over all ways how the color set $f(a)$ can be split among the vertices with label a in the first graph G_{t_1} and in the second graph G_{t_2}.

This recurrence can be computed for all f simultaneously in time $\mathcal{O}^*((2^r)^k)$ by turning it into an appropriate cover product in the min-sum semiring as follows. We interpret the functions of the form $f\colon [k] \to \mathcal{P}([r])$ as subsets of $[k] \times [r]$ in the following way: $S(f) = \{(i, c) : i \in [k], c \in f(i)\}$. Observe that $f_1(a) \cup f_2(a) = f(a)$ for all $a \in [k]$ is equivalent to $S(f_1) \cup S(f_2) = S(f)$. Now, A_t can be considered as a function $\mathcal{P}([k] \times [r]) \to [n]$ and the recurrence of the union case is the $(\min, +)$-cover product of A_{t_1} and A_{t_2}. By [10, Theorem 10.17] we can compute all values of A_t in time $2^{kr}(kr)^{\mathcal{O}(1)} \cdot \mathcal{O}(n \log n \log \log n) = \mathcal{O}^*((2^r)^k)$.

Correctness. We prove the correctness by bottom-up induction along the syntax tree $\mathcal{T}_\mu$. In the base case G_t only consists of the single vertex v and we can either delete v or assign some color to v. Together with the edge case handling, this is implemented by the formula for the base case.

For the relabel case, notice that $G_t = G_{t'}$, $V(G_t^i) = \emptyset$, $V(G_t^j) = V(G_{t'}^i) \cup V(G_{t'}^j)$, and $V(G_t^a) = V(G_{t'}^a)$ for all $a \in [k] \setminus \{i, j\}$. Let f' be a candidate in the recurrence of $A_t[f]$ and $\varphi' \in \mathcal{Q}_{t'}[f']$ be a minimizer in the definition of $A_{t'}[f']$, then we also have that $\varphi' \in \mathcal{Q}_t[f]$ since $\varphi'(V(G_t^j)) \setminus \{\bot\} = (\varphi'(V(G_{t'}^i)) \setminus \{\bot\}) \cup (\varphi'(V(G_{t'}^j)) \setminus \{\bot\}) = f'(i) \cup f'(j) = f(j)$. Hence, the recurrence is an upper bound on $A_t[f]$.

In the other direction, let φ be a minimizer in the definition of $A_t[f]$ and consider f' with $f'(a) = \varphi(V(G_{t'}^a)) \setminus \{\bot\}$ for all $a \in [k]$. Then f' satisfies $f'(i) \cup f'(j) = f(j)$ and $\varphi \in Q_{t'}[f']$, so f' is also considered in the recurrence and the recurrence is a lower bound on $A_t[f]$.

For the join case, notice that for $\varphi' \in \mathcal{Q}_{t'}[f] \supseteq \mathcal{Q}_t[f]$ it holds that $\varphi' \in \mathcal{Q}_t[f]$ if and only if $\varphi'(V(G_{t'}^i)) \cap \varphi'(V(G_{t'}^j)) \subseteq \{\bot\}$.

For the union case, a feasible solution φ of G_t induces feasible solutions φ_1 of G_{t_1} and φ_2 of G_{t_2} such that $\varphi_1(V(G_t^a)) \cup \varphi_2(V(G_t^a)) = \varphi(V(G_t^a))$ for all $a \in [k]$ and vice versa. ◀

This algorithm has a straightforward extension that can also handle polynomially large vertex costs in running time $\mathcal{O}^*((2^r)^k)$. For even larger costs it is not clear how to compute the table entries for the union nodes quickly enough.

6 Conclusion

Our main results are the two lower bounds for Deletion to r-Colorable, which apply also to parameterization by treewidth resp. cliquewidth but use much more restrictive structure; this greatly refines what was known for Odd Cycle Transversal, i.e. $r = 2$, and gives new tight bounds for $r \geq 3$. In particular, beyond the above-mentioned examples, these are further natural problems where a small modulator to a simple graph class (of constant treewidth) is as hard as small treewidth. Surprisingly perhaps, something even stronger holds for clique-width: To get the tight lower bound, a modulator with few (true) twinclasses suffices, i.e., we need neither a sequence of disjoint separators nor complex dense structure. For Dominating Set, only the latter was established: twinclass-cutwidth rather than cliquewidth suffices to take us from base 3 in the running time to base 4.

Such results bring several benefits: (1) Rather than e.g. getting only the isolated result of (conditional) complexity of a problem relative to treewidth, we get a much larger range of input structure that exhibits the same tight complexity. (2) At the same time, by aiming for maximally restricted lower bound structure, we get a much better understanding of what structure makes a given problem hard. This in turn helps to focus efforts at faster algorithms through (even) stronger structural restrictions on the input.

An immediate follow-up question is whether there are improved algorithms for Deletion to r-Colorable when $G - X$ has treewidth less than r; so far, this is known only for Odd Cycle Transversal, but we think such algorithms exist in general. We observe that any construction relying on $(r + 1)$-critical graphs must have treewidth at least r, hence improving upon the treewidth of our construction requires a fundamentally different idea.

Similarly, is there a meaningful restriction of (linear) clique-width, for which Lampis' [42] lower bound for r-Coloring already holds? Much more broadly, what other classes of problems exhibit the same lower bound as for treewidth already relative to deletion distance to a sparse graph class? Are there problems where this jump in complexity happens later, say, for treedepth, for some elimination distance, or only for treewidth/pathwidth? E.g., what is the complexity of Dominating Set relative to deletion distances, and the complexity relative to treedepth may be an interesting stepping stone? Similarly, to what generality do we get the same lower bound as for clique-width already relative to, e.g., twinclass-pathwidth?

References

1 Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.

2 Glencora Borradaile and Hung Le. Optimal dynamic program for r-domination problems over tree decompositions. In Jiong Guo and Danny Hermelin, editors, *11th International Symposium on Parameterized and Exact Computation, IPEC 2016, August 24-26, 2016, Aarhus, Denmark*, volume 63 of *LIPIcs*, pages 8:1–8:23. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.IPEC.2016.8.

3 Jannis Bulian and Anuj Dawar. Graph isomorphism parameterized by elimination distance to bounded degree. *Algorithmica*, 75(2):363–382, 2016. doi:10.1007/s00453-015-0045-3.

4 Jannis Bulian and Anuj Dawar. Fixed-parameter tractable distances to sparse graph classes. *Algorithmica*, 79(1):139–158, 2017. doi:10.1007/s00453-016-0235-7.

5 Yijia Chen and Jörg Flum. Fo-definability of shrub-depth. In Maribel Fernández and Anca Muscholl, editors, *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain*, volume 152 of *LIPIcs*, pages 15:1–15:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CSL.2020.15.

6 Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discret. Appl. Math.*, 101(1-3):77–114, 2000. doi:10.1016/S0166-218X(99)00184-5.

7 Radu Curticapean, Nathan Lindzey, and Jesper Nederlof. A tight lower bound for counting hamiltonian cycles via matrix rank. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1080–1099. SIAM, 2018. doi:10.1137/1.9781611975031.70.

8 Radu Curticapean and Dániel Marx. Tight conditional lower bounds for counting perfect matchings on graphs of bounded treewidth, cliquewidth, and genus. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1650–1669. SIAM, 2016. doi:10.1137/1.9781611974331.ch113.

9 Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as CNF-SAT. *ACM Trans. Algorithms*, 12(3):41:1–41:24, 2016. doi:10.1145/2925416.

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

11 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.

12 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michał Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 150–159. IEEE Computer Society, 2011. doi:10.1109/FOCS.2011.23.

13 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.

14 Matt DeVos, O-joung Kwon, and Sang-il Oum. Branch-depth: Generalizing tree-depth of graphs. *Eur. J. Comb.*, 90:103186, 2020. doi:10.1016/j.ejc.2020.103186.

15 Louis Dublois, Michael Lampis, and Vangelis Th. Paschos. New algorithms for mixed dominating set. *Discret. Math. Theor. Comput. Sci.*, 23(1), 2021. URL: http://dmtcs.episciences.org/7407.

16 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.

17 László Egri, Dániel Marx, and Pawel Rzazewski. Finding list homomorphisms from bounded-treewidth graphs to reflexive graphs: a complete complexity characterization. In Rolf Niedermeier and Brigitte Vallée, editors, *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*, volume 96 of *LIPIcs*, pages 27:1–27:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.STACS.2018.27.

18 Eduard Eiben, Robert Ganian, Thekla Hamm, and O-joung Kwon. Measuring what matters: A hybrid approach to dynamic programming with treewidth. *J. Comput. Syst. Sci.*, 121:57–75, 2021. doi:10.1016/j.jcss.2021.04.005.

19 Eduard Eiben, Robert Ganian, and Stefan Szeider. Meta-kernelization using well-structured modulators. *Discret. Appl. Math.*, 248:153–167, 2018. doi:10.1016/j.dam.2017.09.018.

20 Eduard Eiben, Robert Ganian, and Stefan Szeider. Solving problems on graphs of high rank-width. *Algorithmica*, 80(2):742–771, 2018. doi:10.1007/s00453-017-0290-8.

21 Jacob Focke, Dániel Marx, and Pawel Rzazewski. Counting list homomorphisms from graphs of bounded treewidth: tight complexity bounds. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9–12, 2022*, pages 431–458. SIAM, 2022. doi:10.1137/1.9781611977073.22.

22 Jakub Gajarský and Stephan Kreutzer. Computing shrub-depth decompositions. In Christophe Paul and Markus Bläser, editors, *37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020, March 10-13, 2020, Montpellier, France*, volume 154 of *LIPIcs*, pages 56:1–56:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.STACS.2020.56.

23 Jakub Gajarský, Michael Lampis, and Sebastian Ordyniak. Parameterized algorithms for modular-width. In Gregory Z. Gutin and Stefan Szeider, editors, *Parameterized and Exact Computation – 8th International Symposium, IPEC 2013, Sophia Antipolis, France, September 4-6, 2013, Revised Selected Papers*, volume 8246 of *Lecture Notes in Computer Science*, pages 163–176. Springer, 2013. doi:10.1007/978-3-319-03898-8_15.

24 Robert Ganian, Thekla Hamm, Viktoriia Korchemna, Karolina Okrasa, and Kirill Simonov. The fine-grained complexity of graph homomorphism parameterized by clique-width. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 66:1–66:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.66.

25 Robert Ganian, Petr Hlinený, Jaroslav Nesetril, Jan Obdrzálek, and Patrice Ossona de Mendez. Shrub-depth: Capturing height of dense graphs. *Log. Methods Comput. Sci.*, 15(1), 2019. doi:10.23638/LMCS-15(1:7)2019.

26 Carla Groenland, Isja Mannens, Jesper Nederlof, and Krisztina Szilágyi. Tight bounds for counting colorings and connected edge sets parameterized by cutwidth. In Petra Berenbrink and Benjamin Monmege, editors, *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, March 15-18, 2022, Marseille, France (Virtual Conference)*, volume 219 of *LIPIcs*, pages 36:1–36:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.STACS.2022.36.

27 Falko Hegerfeld and Stefan Kratsch. Towards exact structural thresholds for parameterized complexity. *CoRR*, abs/2107.06111, 2021. arXiv:2107.06111.

28 Petr Hlinený, O-joung Kwon, Jan Obdrzálek, and Sebastian Ordyniak. Tree-depth and vertex-minors. *Eur. J. Comb.*, 56:46–56, 2016. doi:10.1016/j.ejc.2016.03.001.

29 Eva-Maria C. Hols, Stefan Kratsch, and Astrid Pieterse. Elimination distances, blocking sets, and kernels for vertex cover. In Christophe Paul and Markus Bläser, editors, *37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020, March 10-13, 2020, Montpellier, France*, volume 154 of *LIPIcs*, pages 36:1–36:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.STACS.2020.36.

30 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

31 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

32 Yoichi Iwata and Yuichi Yoshida. On the equivalence among problems of bounded width. In Nikhil Bansal and Irene Finocchi, editors, *Algorithms – ESA 2015 – 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, volume 9294 of *Lecture Notes in Computer Science*, pages 754–765. Springer, 2015. doi:10.1007/978-3-662-48350-3_63.

33 Ashwin Jacob, Fahad Panolan, Venkatesh Raman, and Vibha Sahlot. Structural parameterizations with modulator oblivion. *Algorithmica*, 84(8):2335–2357, 2022. doi:10.1007/s00453-022-00971-7.

34 Hugo Jacob, Thomas Bellitto, Oscar Defrain, and Marcin Pilipczuk. Close relatives (of feedback vertex set), revisited. In Petr A. Golovach and Meirav Zehavi, editors, *16th International Symposium on Parameterized and Exact Computation, IPEC 2021, September 8-10, 2021, Lisbon, Portugal*, volume 214 of *LIPIcs*, pages 21:1–21:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.IPEC.2021.21.

35 Lars Jaffke and Bart M. P. Jansen. Fine-grained parameterized complexity analysis of graph coloring problems. In Dimitris Fotakis, Aris Pagourtzis, and Vangelis Th. Paschos, editors, *Algorithms and Complexity – 10th International Conference, CIAC 2017, Athens, Greece, May 24-26, 2017, Proceedings*, volume 10236 of *Lecture Notes in Computer Science*, pages 345–356, 2017. doi:10.1007/978-3-319-57586-5_29.

36 Bart M. P. Jansen, Jari J. H. de Kroon, and Michal Wlodarczyk. Vertex deletion parameterized by elimination distance and even less. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1757–1769. ACM, 2021. doi:10.1145/3406325.3451068.

37 Bart M. P. Jansen and Jesper Nederlof. Computing the chromatic number using graph decompositions via matrix rank. *Theor. Comput. Sci.*, 795:520–539, 2019. doi:10.1016/j.tcs.2019.08.006.

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

39 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.

40 O-joung Kwon, Rose McCarty, Sang-il Oum, and Paul Wollan. Obstructions for bounded shrub-depth and rank-depth. *J. Comb. Theory, Ser. B*, 149:76–91, 2021. doi:10.1016/j.jctb.2021.01.005.

41 Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012. doi:10.1007/s00453-011-9554-x.

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

43 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.

44 Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Trans. Algorithms*, 11(2):15:1–15:31, 2014. doi:10.1145/2566616.

45 Dániel Marx, Govind S. Sankar, and Philipp Schepper. Degrees and gaps: Tight complexity results of general factor problems parameterized by treewidth and cutwidth. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 95:1–95:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.95.

46 Dániel Marx, Govind S. Sankar, and Philipp Schepper. Anti-factor is FPT parameterized by treewidth and list size (but counting is hard). *CoRR*, abs/2110.09369, 2022. To appear at IPEC 2022. arXiv:2110.09369.

47 Stefan Mengel. Parameterized compilation lower bounds for restricted CNF-formulas. In Nadia Creignou and Daniel Le Berre, editors, *Theory and Applications of Satisfiability Testing – SAT 2016 – 19th International Conference, Bordeaux, France, July 5-8, 2016, Proceedings*, volume 9710 of *Lecture Notes in Computer Science*, pages 3–12. Springer, 2016. doi:10.1007/978-3-319-40970-2_1.

48 Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity – Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-27875-4.

49 Karolina Okrasa, Marta Piecyk, and Pawel Rzazewski. Full complexity classification of the list homomorphism problem for bounded-treewidth graphs. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 74:1–74:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ESA.2020.74.

50 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.

51 Daniël Paulusma, Friedrich Slivovsky, and Stefan Szeider. Model counting for CNF formulas of bounded modular treewidth. *Algorithmica*, 76(1):168–194, 2016. doi:10.1007/s00453-015-0030-x.

52 Marta Piecyk and Pawel Rzazewski. Fine-grained complexity of the list homomorphism problem: Feedback vertex set and cutwidth. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference)*, volume 187 of *LIPIcs*, pages 56:1–56:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPIcs.STACS.2021.56`.

53 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`.

A Problem Definitions

VERTEX COVER

Input: An undirected graph $G = (V, E)$ and an integer b.

Question: Is there a set $Y \subseteq V$, $|Y| \leq b$, such that $G - Y$ contains no edges, i.e., $\chi(G - Y) \leq 1$?

ODD CYCLE TRANSVERSAL

Input: An undirected graph $G = (V, E)$ and an integer b.

Question: Is there a set $Y \subseteq V$, $|Y| \leq b$, such that $G - Y$ is bipartite, i.e., $\chi(G - Y) \leq 2$?

DELETION TO r-COLORABLE

Input: An undirected graph $G = (V, E)$ and an integer b.

Question: Is there a set $Y \subseteq V$, $|Y| \leq b$, such that $\chi(G - Y) \leq r$?

SATISFIABILITY

Input: A boolean formula σ in conjunctive normal form.

Question: Is there a satisfying assignment τ for σ?

---

q-SATISFIABILITY

Input: A boolean formula σ in conjunctive normal form with clauses of size at most q.

Question: Is there a satisfying assignment τ for σ?

---

q-HITTING SET

Input: An universe U and a set family $\mathcal{F}$ over U of sets of size at most q and an integer t.

Question: Is there a set $H \subseteq U$, $|H| \leq t$, such that $H \cap S \neq \emptyset$ for all $S \in \mathcal{F}$?

---

r-Coloring

Input: An undirected graph $G = (V, E)$.

Question: Is $\chi(G) \leq r$?

List r-Coloring

Input: An undirected graph $G = (V, E)$, lists $\Lambda(v) \subseteq [r]$ for all $v \in V$.

Question: Is there an r-Coloring $\varphi \colon V \to [r]$ of G such that $\varphi(v) \in \Lambda(v)$ for all $v \in V$?

Maximum Cut

Input: An undirected graph $G = (V, E)$ and an integer b.

Question: Is there a set $Y \subseteq V$, such that $|\delta(Y)| \geq b$?

H-free Deletion

Input: An undirected graph $G = (V, E)$ and an integer b.

Question: Is there a set $Y \subseteq V$, $|Y| \leq b$, such that $G - Y$ is H-free?

Dominating Set

Input: An undirected graph $G = (V, E)$ and an integer b.

Question: Is there a set $X \subseteq V$, $|X| \leq b$, such that $N[X] = V$?

Total Dominating Set

Input: An undirected graph $G = (V, E)$ and an integer b.

Question: Is there a set $X \subseteq V$, $|X| \leq b$, such that $\bigcup_{v \in X} N(v) = V$?

(b, r)-center

Input: An undirected graph $G = (V, E)$ and an integers b and r.

Question: Is there a set $X \subseteq V$, $|X| \leq b$, such that every vertex $v \in V$ is at most at distance r to X?