Fundamental Problems on Bounded-Treewidth Graphs: The Real Source of Hardness

Barış Can Esmer

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
Saarbrücken Graduate School of Computer Science, Saarland Informatics Campus, Germany

Jacob Focke

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Dániel Marx

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Paweł Rzążewski

Warsaw University of Technology, Poland
University of Warsaw, Poland

Abstract

It is known for many algorithmic problems that if a tree decomposition of width t is given in the input, then the problem can be solved with exponential dependence on t. A line of research initiated by Lokshtanov, Marx, and Saurabh [SODA 2011] produced lower bounds showing that in many cases known algorithms already achieve the best possible exponential dependence on t, assuming the Strong Exponential-Time Hypothesis (SETH). The main message of this paper is showing that the same lower bounds can already be obtained in a much more restricted setting: informally, a graph consisting of a block of t vertices connected to components of constant size already has the same hardness as a general tree decomposition of width t.

Formally, a (σ, δ)-*hub* is a set Q of vertices such that every component of Q has size at most σ and is adjacent to at most δ vertices of Q. We explore if the known tight lower bounds parameterized by the width of the given tree decomposition remain valid if we parameterize by the size of the given hub.

- For every $\varepsilon > 0$, there are $\sigma, \delta > 0$ such that Independent Set (equivalently Vertex Cover) cannot be solved in time $(2-\varepsilon)^p \cdot n$, even if a (σ, δ)-hub of size p is given in the input, assuming the SETH. This matches the earlier tight lower bounds parameterized by width of the tree decomposition. Similar tight bounds are obtained for Odd Cycle Transversal, Max Cut, q-Coloring, and edge/vertex deletions versions of q-Coloring.
- For every $\varepsilon > 0$, there are $\sigma, \delta > 0$ such that $\triangle$-Partition cannot be solved in time $(2-\varepsilon)^p \cdot n$, even if a (σ, δ)-hub of size p is given in the input, assuming the Set Cover Conjecture (SCC). In fact, we prove that this statement is *equivalent* to the SCC, thus it is unlikely that this could be proved assuming the SETH.
- For Dominating Set, we can prove a non-tight lower bound ruling out $(2-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$ algorithms, assuming *either* the SETH or the SCC, but this does not match the $3^p \cdot n^{\mathcal{O}(1)}$ upper bound.

Thus our results reveal that, for many problems, the research on lower bounds on the dependence on tree width was never really about tree decompositions, but the real source of hardness comes from a much *simpler* structure.

Additionally, we study if the same lower bounds can be obtained if σ and δ are fixed universal constants (not depending on ε). We show that lower bounds of this form are possible for Max Cut and the edge-deletion version of q-Coloring, under the Max 3-Sat Hypothesis (M3SH). However, no such lower bounds are possible for Independent Set, Odd Cycle Transversal, and the vertex-deletion version of q-Coloring: better than brute force algorithms are possible for every fixed (σ, δ).

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

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).
Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson;
Article No. 34; pp. 34:1–34:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Keywords and phrases Parameterized Complexity, Tight Bounds, Hub, Treewidth, Strong Exponential Time Hypothesis, Vertex Coloring, Vertex Deletion, Edge Deletion, Triangle Packing, Triangle Partition, Set Cover Hypothesis, Dominating Set

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.34

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://doi.org/10.48550/arXiv.2402.07331

1 Introduction

Starting with the work of Lokshtanov, Marx, and Saurabh [24], there is a line of research devoted to giving lower bounds on how the running time of parameterized algorithms can depend on treewidth (or more precisely, on the width of a given tree decomposition) [32, 31, 11, 7, 4, 19, 27, 8, 13, 28, 12]. The goal of this paper is to revisit the fundamental results from [24] to point out that previous work could have considered a *simpler* parameter to obtain *stronger* lower bounds in a more uniform way. Thus, in a sense, this line of research was never really about treewidth; a fact that future work should take into account.

Suppose we want to solve some algorithmic problem on a graph G given with a tree decomposition of width t. For many NP-hard problems, standard dynamic program techniques or meta theorems such as Courcelle's Theorem [5] show that the problem can be solved in time $f(t) \cdot n^{\mathcal{O}(1)}$ for some computable function f [10, Chapter 7]. In many cases, the running time is actually $c^t \cdot n^{\mathcal{O}(1)}$ for some constant $c > 1$, where it is an obvious goal to make the constant as small as possible. A line of work started by Lokshtanov, Marx, and Saurabh [24] provides tight conditional lower bounds for many problems with known $c^t \cdot n^{\mathcal{O}(1)}$-time algorithms. The lower bounds are based on the Strong Exponential-Time Hypothesis, formulated by Impagliazzo, Paturi, and Zane [16, 17].

▶ **Strong Exponential-Time Hypothesis (SETH).** *There is no $\varepsilon > 0$ such that for every k, every n-variable instance of k-SAT can be solved in time $(2-\varepsilon)^n \cdot n^{\mathcal{O}(1)}$.*

The goal of these results is to provide evidence that the base c of the exponent in the best known $c^t \cdot n^{\mathcal{O}(1)}$-time algorithm is optimal: if a $(c-\varepsilon)^t \cdot n^{\mathcal{O}(1)}$-time algorithm exists for any $\varepsilon > 0$, then SETH fails. The following theorem summarizes the basic results obtained by Lokshtanov, Marx, and Saurabh [24].

▶ **Theorem 1.1** ([24]). *If there exists an $\varepsilon > 0$ such that*
1. Independent Set *can be solved in time $(2-\varepsilon)^t \cdot n^{\mathcal{O}(1)}$, or*
2. Dominating Set *can be solved in time $(3-\varepsilon)^t \cdot n^{\mathcal{O}(1)}$, or*
3. Max Cut *can be solved in time $(2-\varepsilon)^t \cdot n^{\mathcal{O}(1)}$, or*
4. Odd Cycle Transversal *can be solved in time $(3-\varepsilon)^t \cdot n^{\mathcal{O}(1)}$, or*
5. *q-*Coloring *can be solved in time $(q-\varepsilon)^t \cdot n^{\mathcal{O}(1)}$ for some $q \geqslant 3$, or*
6. *Triangle Partition can be solved in time $(2-\varepsilon)^t \cdot n^{\mathcal{O}(1)}$,*

on input an n-vertex graph G together with a tree decomposition of width at most t, then the SETH fails.

Already in [24] it is pointed out that many of the lower bounds remain true even in the more restricted setting where the input is not a tree decomposition, but a path decomposition. This raises the following natural questions:
- How much further can we restrict the input and still obtain the same lower bounds?
- What is the *real* structural source of hardness in these results?

In this paper, we show that many of these lower bounds remain true in a much more restricted setting where a block of p vertices is connected to constant-size components. Additionally, we demonstrate that our results are very close to being best possible, as further restrictions of the structure of the graphs allow better algorithms.

We say that a set Q of vertices is a (σ, δ)-*hub* of G if every component of $G - Q$ has at most σ vertices and each such component is adjacent to at most δ vertices of Q in G[1]. Our goal is to prove lower bounds parameterized by the size of a (σ, δ)-hub given in the input, where σ and δ are treated as constants. One can observe that a (σ, δ)-hub of size p in G can be easily turned into a tree decomposition of width less than $p + \sigma$, hence the treewidth of G is at most $p + \sigma$. Therefore, any lower bound parameterized by the size p of hub immediately implies a lower bound parameterized by the width of the given tree decomposition. We systematically go through the list of problems investigated by Lokshtanov, Marx, and Saurabh [24], to see if the same lower bound can be obtained with this parameterization. Our results show that, in most cases, the results remain valid under parameterization by hub size. However, new insights, techniques and arguments are needed; in particular, we require different complexity assumptions for some of the statements.

1.1 Coloring Problems and Relatives

Let us first consider the q-Coloring problem: given a graph G, the task is to find a coloring of the vertices of G with q colors such that adjacent vertices receive different colors. Given a (σ, δ)-hub Q of size p, we can try all possible q-colorings on Q and check if they can be extented to every component of $G - Q$. Assuming σ and δ are constants, this leads to a $q^p \cdot n^{\mathcal{O}(1)}$ algorithm. Our first result shows that this is essentially best possible, assuming the SETH; note that this result immediately implies Theorem 1.1(5).

▶ **Theorem 1.2.** *Let $q \geqslant 3$ be an integer.*

1. *For every $\sigma, \delta \geqslant 1$, q-Coloring on n-vertex graphs can be solved in time $q^p \cdot n^{\mathcal{O}(1)}$ if a (σ, δ)-hub of size p is given in the input.*
2. *For every $\varepsilon > 0$, there exist integers $\sigma, \delta \geqslant 1$ such that if there is an algorithm solving in time $(q - \varepsilon)^p \cdot n^{\mathcal{O}(1)}$ every n-vertex instance of q-Coloring given with a (σ, δ)-hub of size at most p, then the SETH fails.*

The q-ColoringED problem is an edge-deletion optimization version of q-Coloring: given a graph G, the task is to find a set X of edges of minimum size such that $G \setminus X$ has a q-coloring. We show that $q^p \cdot n^{\mathcal{O}(1)}$ running time is essentially optimal for this problem as well.

▶ **Theorem 1.3.** *Let $q \geqslant 2$ be an integer.*

1. *For every $\sigma, \delta \geqslant 1$, q-ColoringED on n-vertex graphs can be solved in time $q^p \cdot n^{\mathcal{O}(1)}$ if a (σ, δ)-hub of size p is given in the input.*
2. *For every $\varepsilon > 0$, there exist integers $\sigma, \delta \geqslant 1$ such that if there is an algorithm solving in time $(q - \varepsilon)^p \cdot n^{\mathcal{O}(1)}$ every n-vertex instance of q-ColoringED given with a (σ, δ)-hub of size at most p, then the SETH fails.*

[1] This notion is related to *component order connectivity*, which is the size of the smallest set Q of vertices such that deleting Q leaves components of size not larger than some predefined constant σ [37, 33, 26, 3, 20, 22, 1, 14, 34, 6, 25]. Our definition has the additional constraint on the neighborhood size of each component. As we often refer to the set Q itself (not only its smallest possible size) and we want to make the constants σ, δ explicit, the terminology (σ, δ)-hub is grammatically more convenient than trying to express the same using component order connectivity.

For $q \geqslant 3$, the lower bound of Theorem 1.2 for q-Coloring immediately implies the same lower bound for the more general problem q-ColoringED. Observe that for $q = 2$, the q-ColoringED problem is equivalent to the Max Cut problem: deleting the minimum number of edges to make the graph bipartite is equivalent to finding a bipartition with the maximum number of edges going between the two classes. Thus the lower bound for Max Cut is needed to complete the proof of Theorem 1.3.

Let us consider now the vertex-deletion version q-ColoringVD, where given a graph G, the task is to find a set X of vertices of minimum size such that $G - X$ has a q-coloring (equivalently, we want to find a partial q-coloring on the maximum number of vertices). For this problem, a brute force approach would need to consider $(q+1)^p$ possibilities on a (σ, δ)-hub of size p: each vertex can receive either one of the q colors, or be deleted.

▶ **Theorem 1.4.** *Let $q \geqslant 1$ be an integer.*

1. *For every $\sigma, \delta \geqslant 1$, q-ColoringVD on n-vertex graphs can be solved in time $(q+1)^p \cdot n^{\mathcal{O}(1)}$ if a (σ, δ)-hub of size p is given in the input.*
2. *For every $\varepsilon > 0$, there exist integers $\sigma, \delta \geqslant 1$ such that if there is an algorithm solving in time $(q+1-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$ every n-vertex instance of q-ColoringVD given with a (σ, δ)-hub of size at most p, then the SETH fails.*

Observe that Vertex Cover is equivalent to 1-ColoringVD and Odd Cycle Transversal is equivalent to 2-ColoringVD. Furthermore, Independent Set and Vertex Cover have the same time complexity (due to the well-known fact that minimum size of a vertex cover plus the maximum size of an independent set is always equal to the number of vertices). Thus the definition of q-ColoringVD gives a convenient unified formulation that includes these fundamental problems.

1.2 Packing Problems

Given a graph G, the Triangle Partition (denoted by $\triangle$-Partition for short) problem asks for a partition of the vertex set into triangles. Triangle Packing (denoted by $\triangle$-Packing) is the more general problem where the task is to find a maximum-size collection of vertex-disjoint triangles. Given a tree decomposition of width t, Theorem 1.1(6) shows that $2^t \cdot n^{\mathcal{O}(1)}$ is essentially the best possible running time. It seems that the same lower bound holds when parameterizing by the size of a hub, but the source of hardness is somehow different. Instead of assuming the SETH, we prove this lower bound under the *Set Cover Conjecture (SCC)* [9, 10]. In the d-Set Cover problem, we are given a universe U of size n and a collection $\mathcal{F}$ of subsets of U, each with size at most d. The task is to find a minimum-size collection of sets whose union covers the universe.

▶ **Set Cover Conjecture (SCC).** *For all $\varepsilon > 0$, there exists $d \geqslant 1$ such that there is no algorithm that solves every $\leqslant d$-Set Cover instance $(U, \mathcal{F})$ in time $(2-\varepsilon)^n \cdot n^{\mathcal{O}(1)}$ where $n = |U|$.*

We actually show that the lower bounds for $\triangle$-Partition/$\triangle$-Packing are *equivalent* to the SCC.

▶ **Theorem 1.5.** *The following three statements are equivalent:*

- *The SCC is true.*
- *For every $\varepsilon > 0$, there are $\sigma, \delta > 0$ such that $\triangle$-Partition on an n-vertex graph cannot be solved in time $(2-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$, even if the input contains a (σ, δ)-hub of size p.*
- *For every $\varepsilon > 0$, there are $\sigma, \delta > 0$ such that $\triangle$-Packing on an n-vertex graph cannot be solved in time $(2-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$, even if the input contains a (σ, δ)-hub of size p.*

Ideally, one would like to prove lower bounds under the more established conjecture: the SETH. However, Theorem 1.5 shows that it is no shortcoming of our technique that we prove the lower bound based on the SCC instead. If we proved statement 2 or 3 under the SETH, then this would prove that the SETH implies the SCC, resolving a longstanding open question.

1.3 Dominating Set

Given an n-vertex graph with a tree decomposition of width t, a minimum dominating set can be computed in time $3^t \cdot n^{\mathcal{O}(1)}$ using an algorithm based on fast subset convolution [35, 36]. By Theorem 1.1 (2), this running time cannot be improved to $(3-\varepsilon)^t \cdot n^{\mathcal{O}(1)}$ for any $\varepsilon > 0$, assuming the SETH. Can we get a $(3-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$ algorithm if a hub of size p is given in the input? We currently have no answer to this question. In fact, we do not even have a good guess whether or not such algorithms should be possible. What we do have are two very simple weaker results that rule out $(2-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$ algorithms, with one proof based on the SETH and the other proof based on the SCC.

▶ **Theorem 1.6.** *For every $\varepsilon > 0$, there are $\sigma, \delta > 0$ such that* Dominating Set *on an n-vertex graph with a (σ, δ)-hub of size p given in the input cannot be solved in time $(2-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$, unless* both *the SETH and the SCC fail.*

Theorem 1.6 suggests that, if there is no $(3-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$ time algorithm for Dominating Set, then perhaps the matching lower bound needs a complexity assumption that is stronger than both the SETH and the SCC.

1.4 Universal Constants for σ and δ?

The lower bounds in Theorems 1.2–1.6 are stated in a somewhat technical form: "for every $\varepsilon > 0$, there are σ and δ such that…". The statements would be simpler and more intuitive if they were formulated in a setting where σ and δ are universal constants, say, 100. Can we prove statements that show, for example, that there is no $(2-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$ algorithm, where p is the size of a $(100, 100)$-hub given in the input? The answer to this question is complicated. For the vertex-deletion problem q-ColoringVD (which includes Vertex Cover and Odd Cycle Transversal) there are actually better than brute force algorithms for fixed constant values of σ and δ.

▶ **Theorem 1.7.** *For every $q \geqslant 3$ and $\sigma, \delta > 0$, there exists $\varepsilon > 0$ with the following property: every instance (G, L) of q-ColoringVD with n vertices, given with a (σ, δ)-hub of size p, can be solved in time $(q+1-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$.*

Thus Theorem 1.7 explains why the formulation of Theorem 1.4 needs to quantify over σ and δ, and cannot be stated for a fixed pair (σ, δ).

On the other hand, for the edge-deletion problem q-ColoringED (which includes Max Cut), we can prove stronger lower bounds where σ and δ are universal constants. However, we need a complexity assumption different from the SETH.

An instance of Max 3-Sat is a CNF formula φ with at most three literals in each clause. We ask for the minimum number of clauses that need to be deleted in order to obtain a satisfiable formula. Equivalently, we look for a valuation of the variables which violates the minimum number of clauses. Clearly, an instance of Max 3-Sat with n variables can be solved in time $2^n \cdot n^{\mathcal{O}(1)}$ by exhaustive search. It is a notorious problem whether this running time can be significantly improved, i.e., whether there exists an $\varepsilon > 0$ such that every n-variable instance of Max 3-Sat can be solved in time $(2-\varepsilon)^n$.

▶ **Max 3-Sat Hypothesis (M3SH).** *There is no $\varepsilon > 0$ such that every n-variable instance of* Max 3-Sat *can be solved in time $(2-\varepsilon)^n \cdot n^{\mathcal{O}(1)}$.*

Under this assumption, we can prove a lower bound where $\delta = 6$ and σ is a constant (depending only on q).

▶ **Theorem 1.8.** *For every $q \geqslant 2$ there is an integer σ such that the following holds. For every $\varepsilon > 0$, no algorithm solves every n-vertex instance of q-*ColoringED *that is given with a $(\sigma, 6)$-hub of size p, in time $(q-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$, unless the M3SH fails.*

For the case $q = 2$ we even show a slight improvement over Theorem 1.8 – in this case it suffices to consider instances with a constant σ and $\delta = 4$.

For $\triangle$-Partition, we do not know if the lower bound of Theorem 1.5, ruling out $(2-\varepsilon)^p \cdot n^{O(1)}$ running time under the SCC, remains valid for some fixed universal σ and δ independent of ε. Note that the proof of Theorem 1.5 provides a reduction from $\triangle$-Partition to d-Set Packing for some d. It is known that d-Set Packing over a universe of size n can be solved in time $(2-\varepsilon)^n \cdot (n+m)^{\mathcal{O}(1)}$ with some $\varepsilon > 0$ depending on d [30, 21, 2]. However, our reduction from $\triangle$-Partition to d-Set Packing chooses d in a way that it cannot be used to reduce the case of a fixed σ and δ to a d-Set Packing problem with fixed d. It seems that we would need to understand if certain generalizations of d-Set Packing can also be solved in time $(2-\varepsilon)^n \cdot (n+m)^{\mathcal{O}(1)}$ for fixed d. The simplest such problem would be the generalization of d-Set Packing where the sets in the input are partitioned into pairs and the solution is allowed to use at most one set from each pair.

1.5 Discussion

Given the amount of attention to algorithms on tree decompositions and the number of nontrivial techniques that were developed to achieve the best known algorithms, it is a natural question to ask if these algorithms are optimal. Even though understanding treewidth is a very natural motivation for this line of research, the actual results turned out to be less related to treewidth than one would assume initially: the lower bounds remain valid even under more restricted conditions. Already the first paper on this topic [24] states the lower bounds in a stronger form, as parameterized by pathwidth or by feedback vertex set number (both of which are bounded below by treewidth). Some other results considered parameters such as the size of a set Q where every component of $G - Q$ is a path [18] or has bounded treewidth [15]. However, our results show that none of these lower bounds got to the fundamental reason why known algorithms on bounded-treewidth graphs cannot be improved: Theorems 1.2–1.5 highlight that these algorithms are best possible already if we consider a much more restricted problem setting where constant-sized gadgets are attached to a set of hub vertices. Moreover, Theorems 1.2 and 1.4 are likely to be best possible: as we have seen, for coloring and its vertex-deletion generalizations, σ and δ cannot be made a constant independent from ε (Theorem 1.7). Therefore, one additional conceptual message of our results is understanding where the hardness of solving problems on bounded-treewidth graphs really stems from, by reaching the arguably most restricted setting in which the lower bounds hold.

The success of Theorems 1.2–1.4 (for coloring problems and relatives) suggests that possibly all the treewidth optimality results could be revisited and the same methodology could be used to strengthen to parameterization by hub size. But the story is more complicated than that. For example, for $\triangle$-Packing, the ground truth appears to be that the lower bound parameterized by width of the tree decomposition can be strengthened to a lower bound parameterized by hub size. However, proving the lower bound parameterized by hub size requires a different proof technique, and we can do it only by assuming the SCC – and for all we know this is an assumption orthogonal to the SETH. In fact, we showed that the lower bound for △-Packing is equivalent to the SCC, making it unlikely that a simple proof based on the SETH exists. For Dominating Set, we currently do not know how to obtain tight bounds, highlighting that it is far from granted that all results parameterized by width of tree decomposition can be easily turned into lower bounds parameterized by hub size.

Another important aspect of our results is the delicate way they have to be formulated, with the values of σ and δ depending on ε. Theorem 1.8 shows that in some cases it is possible to prove a stronger bound where σ and δ are universal constants, but this comes at a cost of choosing a different complexity assumption (M3SH). Thus, there is a tradeoff between the choice of the complexity assumption and the strength of the lower bound. In general, it seems that the choice of complexity assumption can play a crucial role in these kind of lower bounds parameterized by hub size. This has to be contrasted with the case of parameterization by the width of the tree decomposition, where the known lower bounds are obtained from the SETH (or its counting version).

It would be natural to try to obtain lower bounds parameterized by hub size for other algorithmic problems as well. The lower bounds obtained in this paper for various fundamental problems can serve as a starting point for such further results. Concerning the problems studied in this paper, we leave two main open questions:

- For Dominating Set, can we improve the lower bound of Theorem 1.6 to rule out $(3-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$ algorithms, under some reasonable assumption? Or is there perhaps an algorithm beating this bound?
- For △-Partition/△-Packing, can we improve the lower bound of Theorem 1.5 such that σ and δ are universal constants? Or is it true perhaps that for every fixed σ and δ, there is an algorithm solving these problems in time $(2-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$ for some $\varepsilon > 0$?

2 Technical Overview

In this section, we overview some of the most important technical ideas in our results.

2.1 q-Coloring

The algorithmic statement in Theorem 1.2 is easily obtained via a simple branching procedure. For the hardness part, we use a lower bound of Lampis [23] for constraint satisfaction problems (CSP) as a starting point: for any $\varepsilon > 0$ and integer d, there is an integer r such that there is no algorithm solving CSP on n variables of domain size d and r-ary constraints in time $(d-\varepsilon)^n$. Therefore, to prove Theorem 1.2 (2), we give a reduction that, given an n-variable CSP instance where the variables are over $[q]$ and the arity of constraints is some constant r, creates an instance of q-Coloring having a hub of size roughly n.

First, we introduce a set of n main vertices in the hub, representing the variables of the CSP instance. We would like to represent each r-ary constraint with a gadget that is attached to a set S of r vertices. We will first allow our gadgets to use lists that specify to which colors certain vertices are allowed to be mapped. In a second step we then remove these lists.

A bit more formally, we say that an *r-ary q-gadget* is a graph J together with a list assignment $L\colon V(J) \to 2^{[q]}$ and r distinguished vertices $\mathbf{x} = (z_1, \ldots, z_r)$ from J. The vertices $z_1, \ldots, z_r$ are called *portals*. A *list coloring* of (J, L) is an assignment $\varphi\colon V(J) \to [q]$ that respects the lists L, i.e., with $\varphi(v) \in L(v)$ for all $v \in V(J)$.

A construction by Jaffke and Jansen [18] gives a gadget that enforces that a set of vertices forbids one prescribed coloring. We use this statement to construct a gadget extends precisely the set of colorings that are allowed according to some relation.

▶ **Proposition 2.1.** *Let $q \geqslant 3$ and $r \geqslant 1$ be integers, and let $R \subseteq [q]^r$ be a relation. Then there exists an r-ary q-gadget $\mathcal{F} = (F, L, (z_1, \ldots, z_r))$ such that*

- *the list of every vertex is contained in $[q]$,*
- *for each $i \in r$, it holds that $L(z_i) = [q]$,*
- *$\{z_1, \ldots, z_r\}$ is an independent set,*
- *for any $\psi : \{z_1, \ldots, z_r\} \to [q]$, coloring vertices $z_1, \ldots, z_r$ according to ψ can be extended to a list coloring of (F, L) if and only if $(\psi(z_1), \ldots, \psi(z_r)) \in R$.*

Then, by introducing one gadget per constraint and attaching it to the vertices of the hub, from the q^n possible behaviors of the hub vertices, only those can be extended to the gadgets that correspond to a satisfying assignment of the CSP instance. Note that gadgets are allowed to use lists and they model the relational constraints using list colorings. So the final step to obtain a reduction to q-Coloring is to remove these lists. This can be done using a standard construction, where a central clique of size q is used to model the q colors, and a vertex v of the graph is adjacent to the ith vertex of the clique, whenever $i \notin L(v)$.

2.2 Vertex Deletion to q-Coloring

Similarly to q-Coloring, the algorithmic statement in Theorem 1.4 is easily obtained via a simple branching procedure. However, for q-ColoringVD, we need to consider $q + 1$ possibilities at each vertex: assigning to it one of the q colors, or deleting it. This leads to the running time $(q + 1)^p \cdot n^{\mathcal{O}(1)}$. The hardness proof is also similar, but this time we have to give a reduction that, given an n-variable CSP instance where the variables are over $[q+1]$ and the arity of constraints is some constant r, creates an instance of q-ColoringVD having a hub of size roughly n. Intuitively, we are using deletion as the $(q + 1)$-st color: the $(q+1)^n$ possibilities for these vertices in the q-ColoringVD problem (coloring with q colors + deletion) correspond to the $(q + 1)^n$ possible assignments of the CSP instance. To enforce this interpretation, we attach to these vertices small gadgets representing each constraint. We attach a large number of copies of each such gadget, which means that it makes no sense for an optimum solution to delete vertices from these gadgets and hence deletions occur only in the hub. This means that we can treat the vertices of the gadgets as “undeletable”.

We would like to use again the construction from Proposition 2.1 to create gadgets that enforce that a set of vertices has one of the prescribed colorings/deletions. A gadget can force the deletion of a vertex if its neighbors are colored using all q colors. However, there is a fundamental limitation of this technique: deleting a vertex is always better than coloring it. That is, a gadget cannot really force a set S of vertices to the color “red”: from the viewpoint of the gadget, deleting some of them and coloring the rest red is equally good. In other words, it is not true that every relation $R \subseteq [q + 1]^r$ can be represented by a gadget that allows only these combinations of q colors + deletion on a set S of r vertices.

To get around this limitation, we use a grouping technique to have control over how many vertices are deleted. Let us divide the n variables into $M = n/b$ blocks $B_1, \ldots, B_M$ of size b each. Let us guess the number f_i of variables in B_i that receive the value $q + 1$ in a hypothetical solution; that is, we expect f_i deletions in block B_i of central vertices. Instead of just attaching a gadget to a set S of at most r vertices, now each gadget is attached to the at most r blocks containing S. Besides ensuring a combination of values on S that satisfies the constraint, the gadget also ensures that each block B_i it is attached to has at least the guessed number f_i of deletions. This way, if we have a solution with exactly $\sum_{i=1}^{M} f_i$ deletions, then we know that it has exactly f_i deletions in the i-th block. Therefore, if a gadget forces the deletion of f_i vertices of B_i and forces a coloring on the remaining vertices of B_i, then we know that that block has exactly this behavior in the solution.

2.3 Edge Deletion to q-Coloring

Let us turn our attention to the edge-deletion version now. Similarly to the vertex-deletion version, the algorithmic results are simple, thus we discuss only the hardness proofs here. As starting point for all our reductions, we use a CSP problem with domain size q that naturally generalizes Max 3-Sat: the task is to find an assignment of variables that satisfies the maximum number of constraints. For $q = 2$, the hardness of this problem follows from the SETH and the M3SH. For $q \geqslant 3$, we prove a new tight lower bound based on M3SH.

For $q \geqslant 3$, the lower bound of Theorem 1.3 (hardness of q-ColoringED under the SETH) already follows from our result for *finding* a q-coloring *without deletions* (Theorem 1.2). So, in order to complete the proof of Theorem 1.3, we give a reduction from the CSP problem with $q = 2$ to 2-ColoringED (i.e., Max Cut), which shows hardness under SETH. As the gadgets of Proposition 2.1 work only for $q \geqslant 3$, we need to design new gadgets using only 2 colors for this case.

The same reduction can be used to establish the lower bound from Theorem 1.8 (hardness of q-ColoringED under the M3SH) in the $q = 2$ case. For the $q \geqslant 3$ case, we present a reduction from the CSP problem with domain size q to q-ColoringED. Here we can once again use the gadgets from Proposition 2.1.

In all cases, as the gadgets we design may use lists, we establish respective lower bounds for the list coloring problem on the way. In a second step, we then show how to remove the lists.

Max CSP – Hardness under the M3SH

For some positive integers d and r, we define Max (d,r)-CSP: Given v variables over a q-element domain and a set of n relational constraints of arity 3, the task is to find an assignment of the variables such that the maximum number of constraints are satisfied. The problem can be solved in time $q^v \cdot n^{\mathcal{O}(1)}$ by brute force. For $q = 2$, the problem is clearly a generalization of Max 3-Sat, hence the M3SH immediately implies that there is no $(q-\varepsilon)^v \cdot n^{\mathcal{O}(1)}$ algorithm for any $\varepsilon > 0$. We show that the M3SH actually implies this for any $q \geqslant 2$. This might also be a helpful tool for future work.

▶ **Theorem 2.2.** *For $d \geqslant 2$ and any $r \geqslant 3$, there is no algorithm solving every n-variable instance of Max (d,r)-CSP in time $(d-\varepsilon)^n \cdot n^{\mathcal{O}(1)}$ for $\varepsilon > 0$, unless the M3SH fails.*

In order to show Theorem 2.2, if q is a power of 2, then a simple grouping argument works: for example, if $q = 2^4 = 16$, then each variable of the CSP instance can represent 4 variables of the Max 3-Sat instance, and hence it is clear that a $(q-\varepsilon)^v$ algorithm would imply a $(2-\varepsilon)^{4v}$ algorithm for a Max 3-Sat instance with $4v$ variables.

The argument is not that simple if q is not a power of 2, say $q = 15$. Then a variable of that CSP instance cannot represent all 16 possibilities of 4 variables of the Max 3-Sat instance, and using it to represent only 3 variables would be wasteful. We cannot use the usual trick of grouping the CSP variables such that each group together represents a group of Max 3-Sat variables: then each constraint representing a clause would need to involve not only 3 variables, but 3 blocks of variables, making δ larger than 3. Instead, for each block of 4 variables of the Max 3-Sat instance, we randomly choose 15 out of the 16 possible assignments, and use a single variable of the CSP instance to represent these possibilities. An optimum solution of a $4v$-variable MAX 3-SAT instance "survives" this random selection with probability $(15/16)^v$. Thus a $(15-\varepsilon)^v \cdot n^{\mathcal{O}(1)}$ time algorithm for the CSP problem would give a randomized $(16/15)^v \cdot (15-\varepsilon)^v \cdot n^{\mathcal{O}(1)} = (16-\varepsilon')^v \cdot n^{\mathcal{O}(1)} = (2-\varepsilon'')^{4v} \cdot n^{\mathcal{O}(1)}$ time algorithm for MAX 3-SAT, violating (a randomized version of) the M3SH. Furthermore, we show in the full version that the argument can be derandomized using the logarithmic integrality gap between integer and fractional covers in hypergraphs.

Realizing Relations using Lists

Recall that an r-ary q-gadget is a graph with lists in $[q]$ and r specified portal vertices. For our hardness proofs, we reduce from MAX (q,r)-CSP, and we use gadgets to "model" the relations in $[q]^r$. We say that an r-ary q-gadget *realizes* a relation $R \in [q]^r$ if there is an integer k such that (1) for each $\mathbf{d} \in R$, if the portals are colored according to $\mathbf{d}$, then it requires precisely k edge deletions to extend this to a full list coloring of the gadget, and (2) extending a state that is *not* in R requires strictly more than k edge deletions. We say that such a gadget 1-realizes R, if for each state outside of R it takes precisely $k+1$ edge deletions to extend this state. So, this is a stronger notion in the sense that now the violation cost is the same for all tuples outside of R. Moreover, with a 1-realizer in hand, by identifying copies of this gadget with the same portal vertices one can freely adjust the precise violation cost – this works as long as the portals form an independent set and therefore no multiedges are introduced in the copying process.

For our treatment of the case $q \geqslant 3$, we again use Proposition 2.1 to show that arbitrary relations over a domain of size q can be realized. As Proposition 2.1 is for the decision problem without deletions, it does not help for the case $q = 2$, i.e., for MAX CUT/2-COLORING. In this case, we need a different approach to show that every relation over a domain of size 2 can be realized. For 2-COLORING, a single edge is essentially a "Not Equals"-gadget as the endpoints have to take different colors or otherwise the edge needs to be deleted. Starting from this, we show how to model OR-relations of any arity. With these building blocks we then obtain the following result.

▶ **Theorem 2.3.** *For each $r \geqslant 1$, and $R \subseteq [2]^r$, there is an r-ary 2-gadget that 1-realizes R.*

Removing the Lists

Note that gadgets may use lists and therefore, on the way, we first obtain the following lower bounds for the respective list coloring problems.

▶ **Theorem 2.4.** *For every $q \geqslant 2$ and $\varepsilon > 0$, there are integers σ and δ such that if an algorithm solves in time $(q-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$ every n-vertex instance of LIST-q-COLORINGED that is given with a (σ, δ)-hub of size p, then the SETH fails.*

▶ **Theorem 2.5.** *For every $q \geqslant 2$, there is a constant σ_q such that, for every $\varepsilon > 0$, if an algorithm solves in time $(q-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$ every n-vertex instance of LIST-q-COLORINGED that is given with a $(\sigma_q, 3)$-hub of size p, then the M3SH fails.*

In a second step, we show how to remove the lists by adding some additional object of size roughly q (a central vertex or a q-clique for $q = 2$ or $q \geqslant 3$, respectively). This addition is then considered to be part of the hub, thereby increasing the size of the hub by some constant. However, this modification means that for the other gadgets the number of neighbors in the hub increases slightly. This is irrelevant for the SETH-based lower bound, but it leads to a slight increase in the universal constant δ that we obtain for our M3SH-based lower bounds for the coloring problems without lists.

2.4 Covering, Packing, and Partitioning

Theorem 1.5 gives lower bounds for $\triangle$-PARTITION and $\triangle$-PACKING based on the Set Cover Conjecture. This hypothesis was formulated in terms of the d-SET COVER problem. For our purposes, it is convenient to consider slightly different covering/partitioning problems. To facilitate our reductions and as a tool for future reductions of this type, we establish equivalences between eight different covering type problems. Before we make this more formal in Theorem 2.6, let us briefly introduce the corresponding problems.

First, we use $=d$-SET COVER and $\leqslant d$-SET COVER to distinguish between the problem for which the sets have size exactly d or at most d, respectively. For $\triangle$-PARTITION, it is more natural to start a reduction from the partitioning problems $=d$-SET PARTITION or $\leqslant d$-SET PARTITION, in which the task is to find pairwise disjoint sets that cover the universe. The $\leqslant d$-SET PARTITION problem can be considered as a decision problem. However, we can also consider the corresponding optimization problem in which the task is to minimize the number of selected sets, and we use $\leqslant d$-SET PARTITION (#SETS) to denote this problem. Further variants are the optimization problems $=d$-SET PACKING (#SETS) and $\leqslant d$-SET PACKING (#SETS), in which we need to select the maximum number of pairwise disjoint sets. For $\leqslant d$-SET PACKING, an equally natural goal is to maximize the total size of the selected sets (for $= d$-SET PACKING, this is of course equivalent to maximizing the number of selected sets). So we use $\leqslant d$-SET PACKING (UNION) to denote the packing problem in which the union/total size of the selected sets is maximized.

Given the large number of variants of d-SET COVER, one may wonder how they are related to each other. In particular, does the SCC imply lower bounds for these variants? There are obvious reductions between some of these problems (e.g., from $= d$-SET COVER to $\leqslant d$-SET COVER) and there are also reductions that are not so straightforward. We fully clarify this question by showing that choosing *any* of these problems in the definition of the SCC leads to an equivalent statement. Thus in our proofs to follow we can choose whichever form is most convenient for us. Knowing this equivalence could prove useful for future work as well.

▶ **Theorem 2.6.** *Suppose that for one of the problems below, it is true that for every $\varepsilon > 0$, there is an integer d such that the problem cannot be solved in time $(2-\varepsilon)^n \cdot n^{\mathcal{O}(1)}$, where n is the size of the universe. Then this holds for all the other problems as well. In particular, any of these statements is equivalent to the SCC.*

1. *$=d$-SET COVER*
2. *$=d$-SET PARTITION*
3. *$=d$-SET PACKING (#SETS)*
4. *$\leqslant d$-SET COVER*
5. *$\leqslant d$-SET PARTITION*
6. *$\leqslant d$-SET PARTITION (#SETS)*
7. *$\leqslant d$-SET PACKING (#SETS)*
8. *$\leqslant d$-SET PACKING (UNION)*

To make the statements about relationships between the problems from the list in Theorem 2.6 more concise, it will be convenient to introduce some shorthand notation. Let $\mathbb{A} = \{A_d\}_{d\geqslant 1}$ and $\mathbb{B} = \{B_d\}_{d\geqslant 1}$ be two families of problems where A_d and B_d belong to the list in Theorem 2.6. To shorten notation, we speak of an *n-element instance* if the universe U of an instance has size n. We say that $\mathbb{A}$ is 2^n-hard if the following lower bound holds

> For each $\varepsilon > 0$ there is some $d \geqslant 1$ such that no algorithm solves A_d on all n-element instances in time $(2-\varepsilon)^n \cdot n^{\mathcal{O}(1)}$.

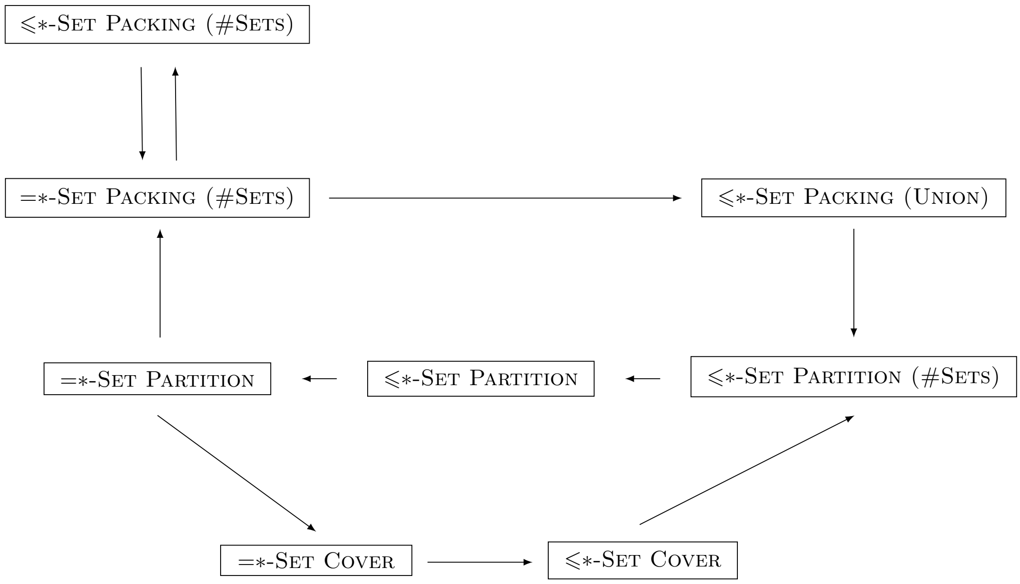

Figure 1 An overview of the proof of Theorem 2.6. An arrow from $\mathcal{A}$ to $\mathcal{B}$ indicates an implication stating that if $\mathcal{A}$ is 2^n-hard then $\mathcal{B}$ is 2^n-hard as well. The details are in the full version.

Using this language, the SCC states that $\{\leqslant d\text{-Set Cover}\}_{d\geqslant 1}$ is 2^n-hard. To establish Theorem 2.6, we show reductions, stating that if $\mathcal{A}$ is 2^n-hard then $\mathcal{B}$ is 2^n-hard as well. Spelled out this means:

Suppose for each $\varepsilon > 0$ there is some $d \geqslant 1$ such that no algorithm solves A_d on all n-element instances in time $(2-\varepsilon)^n \cdot n^{\mathcal{O}(1)}$.

Then, for each $\varepsilon > 0$ there is some $d' \geqslant 1$ such that no algorithm solves $B_{d'}$ on all n-element instances in time $(2-\varepsilon)^n \cdot n^{\mathcal{O}(1)}$.

This shows that this is really a relationship between two classes of problems, and not necessarily a relationship between A_d and B_d for the same value d. To make this distinction explicit, we write $=\ast$-Set Cover if we refer to the class of problems $\{=d\text{-Set Cover}\}_{d\geqslant 1}$. We use analogous notation for the other problems on the list. For example, a simple observation is that if $=\ast$-Set Cover is 2^n-hard then so is $\leqslant\ast$-Set Cover as the latter is a generalization of the former. The reductions we use to prove Theorem 2.6 are illustrated in Figure 1.

2.5 Triangle Partition and Triangle Packing

Now let us discuss the proof of Theorem 1.5 that can be found in the full version. The proof consists of two main steps: (1) a reduction from $=\ast$-Set Partition to $\triangle$-Partition, and (2) a reduction from $\triangle$-Packing to $\leqslant\ast$-Set Packing (#Sets) (see Figure 2). Recall that by Theorem 2.6, assuming the SCC, all of $=\ast$-Set Partition, $\leqslant\ast$-Set Packing (#Sets), and $\leqslant\ast$-Set Cover are 2^n-hard. Finally, $\triangle$-Partition trivially reduces to $\triangle$-Packing, so indeed, the statements in Theorem 1.5 are equivalent.

Reducing Set Partition to $\triangle$-Partition

We start with step (1), i.e., reducing an instance $(U, \mathcal{F})$ of $=d$-Set Partition to an equivalent instance G of $\triangle$-Partition. With a simple technical trick we can ensure that d is divisible by 3.

Figure 2 An overview of the reductions in the proof of Theorem 1.5. The two dashed arrows refer to 2^n-hardness reductions from Theorem 2.6. To establish these two connections, note that we actually utilize almost all the reductions shown in Figure 1. The arrows annotated with (1) and (2) refer to another two reductions proved in the full version.

The main building block used in the reduction is the so-called $\triangle$-eq gadget. For fixed d, it is a graph with d designated vertices called *portals*. The gadget essentially has exactly two triangle packings that cover all non-portal vertices:

- one that also covers *all portals* (i.e., is actually a triangle partition), and
- one that covers *no portal*.

Now the construction of G is simple: we introduce the set Q containing one vertex for each element of U, and for each set $S \in \mathcal{F}$ we introduce a copy of the $\triangle$-eq gadget whose portals represent elements of S and are identified with corresponding vertices from Q. It is straightforward to verify that there is $\mathcal{F}' \subseteq \mathcal{F}$ that partitions U if and only if G has a triangle partition: the sets from $\mathcal{F}'$ correspond to $\triangle$-eq gadgets whose non-portal vertices are covered in the first way. Note that Q is a (σ, d)-hub of G, where σ is the number of vertices of the $\triangle$-eq gadget, i.e., is a constant that depends only on d.

Reducing $\triangle$-Packing to Set Packing.

Now let us consider a graph G given with a (σ, δ)-hub Q of size p, and an integer t. We will show that a hypothetical fast algorithm for $\leqslant d$-Set Packing (#Sets) can be used to determine whether G has a triangle packing of size at least t.

For simplicity of exposition, assume that G has no triangles contained in Q; dealing with such triangles is not difficult but would complicate the notation. We say that a component C of $G - Q$ is *active* in some triangle packing Π if there is a triangle in Π that intersects both C and Q. Note that for any triangle packing there are at most p active components.

We would like to guess components that are active for some (unknown) solution Π. However, this results in too many branches. We deal with it by employing color-coding and reducing the problem to its auxiliary *precolored variant*. Suppose for a moment that we are given a coloring ψ of components of $G - Q$ into p/c colors, where c is a large constant, with a promise that at most c components in each color are active in Π.

For a color $i \in [p/c]$, let $\mathcal{C}_i$ denote the set of components of $G - Q$ colored i by ψ. The *contribution* of the color i to Π is the number of triangles that intersect vertices in components from $\mathcal{C}_i$. Note that the size of Π is the sum of contributions of all color (since we assumed that there are no triangles contained in Q). What can be said about the contribution of i? Certainly picking a maximum triangle packing in the graph consisting only of components from $\mathcal{C}_i$ is a lower bound. Let X_i denote the number of triangles in such a triangle packing and note that X_i can be computed in polynomial time as each component of $G - Q$ is of constant size. Moreover, for each active component $C \in \mathcal{C}_i$, there are at most σ triangles that intersect both C and Q (as each of them has to use a distinct vertex from C). As, by the promise on ψ, there are at most c active components in $\mathcal{C}_i$, we observe that the contribution of i is at most $X_i + c\sigma$. We exhaustively guess the contribution of each color by guessing the offset q_i against X_i; it gives a constant number of options per color. We reject guesses where the total contribution of all colors, i.e., the number of all triangles packed, is less than t.

For each color i, we enumerate all sets $S \subseteq Q$ that are candidates for these vertices of Q that form triangles with vertices from components of $\mathcal{C}_i$; call such sets *i-valid*. An i-valid set S must satisfy the following two conditions. First, the size of S is at most $2c\sigma$, as there are at most $c\sigma$ vertices in active components from $\mathcal{C}_i$ and each such vertex belongs to a triangle with at most two vertices from Q. Second, there exists a triangle packing Π_S in the graph induced by S together with components of $\mathcal{C}_i$ such that

- at most c elements from $\mathcal{C}_i$ are active in Π_S (this follows from the promise on ψ), and
- the number of triangles in Π_S is at least $X_i + q_i$ (by our guess of q_i).

It is not difficult to verify that i-valid sets can be enumerated in polynomial time, where the degree of the polynomial depends on c and σ.

Now we are ready to construct an instance $(U, \mathcal{F}, p/c)$ of $\leqslant d$-Set Packing (#Sets). The universe U is $Q \cup \{a_i \mid i \in [p/c]\}$, i.e., it consists of the hub of G and one extra vertex per color. For each i-valid set S, we include in $\mathcal{F}$ the set $S \cup \{a_i\}$. Again, one can verify that $\mathcal{F}$ contains p/c pairwise disjoint sets if and only if G has a packing of t triangles that agree both with ψ and with the guessed values of q_i's.

By adjusting c, we can ensure that the whole algorithm works in time $(2-\varepsilon')^p \cdot |V(G)|^{\mathcal{O}(1)}$, for some $\varepsilon' > 0$, provided that we have a fast algorithm for $\leqslant d$-Set Packing (#Sets).

The only thing left is to argue how we obtain the coloring ψ satisfying the promise. Here we use *splitters* introduced by Naor, Schulman, and Srinivasan [29]. Informally, a splitter is a family of colorings of a "large set" $\mathcal{X}$, such that for each "small subset" $\mathcal{Y} \subseteq \mathcal{X}$ there is a coloring that splits $\mathcal{Y}$ evenly. In our setting, the "large set" $\mathcal{X}$ is the set of all components of $G - Q$ and the "small subset" $\mathcal{Y}$ is the set of all active components with respect to some fixed (but unknown) solution; recall that there are at most p such active components. Since our colorings use p/c colors, we are sure that there is some ψ for which at most $\frac{p}{p/c} = c$ components in each color are active. Calling the result of Naor, Schulman, and Srinivasan [29], we can find a small splitter Ψ, and then just exhaustively try every coloring $\psi \in \Psi$. Again, carefully adjusting the constants, we can ensure that the overall running time is $(2-\varepsilon)^p \cdot |V(G)|^{\mathcal{O}(1)}$, for some $\varepsilon > 0$.

References

1 Jørgen Bang-Jensen, Eduard Eiben, Gregory Z. Gutin, Magnus Wahlström, and Anders Yeo. Component order connectivity in directed graphs. *Algorithmica*, 84(9):2767–2784, 2022. doi:10.1007/s00453-022-01004-z.

2 Andreas Björklund. Exact Covers via Determinants. In Jean-Yves Marion and Thomas Schwentick, editors, *27th International Symposium on Theoretical Aspects of Computer Science*, volume 5 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 95–106, Dagstuhl, Germany, 2010. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2010.2447.

3 F. Boesch, D. Gross, and C. Suffel. Component order connectivity. In *Proceedings of the Twenty-ninth Southeastern International Conference on Combinatorics, Graph Theory and Computing (Boca Raton, FL, 1998)*, volume 131, pages 145–155, 1998.

4 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 für Informatik, 2016. doi:10.4230/LIPIcs.IPEC.2016.8.

5 Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

6 Christophe Crespelle, Pål Grønås Drange, Fedor V. Fomin, and Petr A. Golovach. A survey of parameterized algorithms and the complexity of edge modification. *Comput. Sci. Rev.*, 48:100556, 2023. doi:10.1016/j.cosrev.2023.100556.

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 Transactions on Algorithms*, 12(3):41:1–41:24, May 2016. doi:10.1145/2925416.

10 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.

11 László Egri, Dániel Marx, and Paweł Rzążewski. 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.

12 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 Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3664–3683. SIAM, 2023. doi:10.1137/1.9781611977554.ch140.

13 Jacob Focke, Dániel Marx, and Paweł Rzążewski. 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.

14 Daniel Gross, L. William Kazmierczak, John T. Saccoman, Charles L. Suffel, and Antonius Suhartomo. On component order edge connectivity of a complete bipartite graph. *Ars Comb.*, 112:433–448, 2013.

15 Falko Hegerfeld and Stefan Kratsch. Towards exact structural thresholds for parameterized complexity. In Holger Dell and Jesper Nederlof, editors, *17th International Symposium on Parameterized and Exact Computation, IPEC 2022, September 7-9, 2022, Potsdam, Germany*, volume 249 of *LIPIcs*, pages 17:1–17:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.IPEC.2022.17.

16 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.

17 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.

18 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.

19 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.

20 Lawrence William Kazmierczak. *On the relationship between connectivity and component order connectivity*. ProQuest LLC, Ann Arbor, MI, 2003. Thesis (Ph.D.)–Stevens Institute of Technology. URL: http://gateway.proquest.com/openurl?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:dissertation&res_dat=xri:pqdiss&rft_dat=xri:pqdiss:3088817.

21 Mikko Koivisto. Partitioning into sets of bounded cardinality. In Jianer Chen and Fedor V. Fomin, editors, *Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers*, volume 5917 of *Lecture Notes in Computer Science*, pages 258–263. Springer, 2009. doi:10.1007/978-3-642-11269-0_21.

22 Mithilesh Kumar and Daniel Lokshtanov. A 2lk kernel for l-component order connectivity. 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 20:1–20:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.IPEC.2016.20.

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

24 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.

25 Daniel Lokshtanov, Pranabendu Misra, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Fpt-approximation for FPT problems. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 199–218. SIAM, 2021. doi:10.1137/1.9781611976465.14.

26 Dániel Marx, Pranabendu Misra, Daniel Neuen, and Prafullkumar Tale. A framework for parameterized subexponential algorithms for generalized cycle hitting problems on planar graphs. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2085–2127. [Society for Industrial and Applied Mathematics (SIAM)], Philadelphia, PA, 2022. doi:10.1137/1.9781611977073.83.

27 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.

28 Dániel Marx, Govind S. Sankar, and Philipp Schepper. Anti-factor is FPT parameterized by treewidth and list size (but counting is hard). In Holger Dell and Jesper Nederlof, editors, *17th International Symposium on Parameterized and Exact Computation, IPEC 2022, September 7-9, 2022, Potsdam, Germany*, volume 249 of *LIPIcs*, pages 22:1–22:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.IPEC.2022.22.

29 Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 182–191. IEEE Computer Society, 1995. doi:10.1109/SFCS.1995.492475.

30 Jesper Nederlof. Finding Large Set Covers Faster via the Representation Method. In Piotr Sankowski and Christos Zaroliagis, editors, *24th Annual European Symposium on Algorithms (ESA 2016)*, volume 57 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 69:1–69:15, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ESA.2016.69`.

31 Karolina Okrasa, Marta Piecyk, and Paweł Rzążewski. 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`.

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

33 Masataka Shirahashi and Naoyuki Kamiyama. Kernelization algorithms for a generalization of the component order connectivity problem. *J. Oper. Res. Soc. Japan*, 66(2):112–129, 2023.

34 Dekel Tsur. Faster parameterized algorithms for two vertex deletion problems. *Theoretical Computer Science*, 940:112–123, 2023. `doi:10.1016/j.tcs.2022.10.044`.

35 Johan M. M. van Rooij. Fast algorithms for join operations on tree decompositions. In Fedor V. Fomin, Stefan Kratsch, and Erik Jan van Leeuwen, editors, *Treewidth, Kernels, and Algorithms - Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, volume 12160 of *Lecture Notes in Computer Science*, pages 262–297. Springer, 2020. `doi:10.1007/978-3-030-42071-0_18`.

36 Johan M. M. van Rooij, Hans L. Bodlaender, and Peter Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In Amos Fiat and Peter Sanders, editors, *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, volume 5757 of *Lecture Notes in Computer Science*, pages 566–577. Springer, 2009. `doi:10.1007/978-3-642-04128-0_51`.

37 M. Yatauro. Component order connectivity and vertex degrees. *Congr. Numer.*, 220:195–205, 2014.