Optimal Data Reduction for Graph Coloring Using Low-Degree Polynomials*

Bart M. P. Jansen[1] **and Astrid Pieterse**[2]

1 **Eindhoven University of Technology**
P.O. Box 513, Eindhoven, The Netherlands
`b.m.p.jansen@tue.nl`

2 **Eindhoven University of Technology**
P.O. Box 513, Eindhoven, The Netherlands
`a.pieterse@tue.nl`

Abstract

The theory of kernelization can be used to rigorously analyze data reduction for graph coloring problems. Here, the aim is to reduce a q-Coloring input to an equivalent but smaller input whose size is provably bounded in terms of structural properties, such as the size of a minimum vertex cover. In this paper we settle two open problems about data reduction for q-Coloring. First, we use a recent technique of finding redundant constraints by representing them as low-degree polynomials, to obtain a kernel of bitsize $\mathcal{O}(k^{q-1} \log k)$ for q-Coloring parameterized by Vertex Cover for any $q \geq 3$. This size bound is optimal up to $k^{o(1)}$ factors assuming $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$, and improves on the previous-best kernel of size $\mathcal{O}(k^q)$. Our second result shows that 3-Coloring does not admit non-trivial sparsification: assuming $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$, the parameterization by the number of vertices n admits no (generalized) kernel of size $\mathcal{O}(n^{2-\varepsilon})$ for any $\varepsilon > 0$. Previously, such a lower bound was only known for coloring with $q \geq 4$ colors.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory

Keywords and phrases graph coloring, kernelization, sparsification

Digital Object Identifier 10.4230/LIPIcs.IPEC.2017.22

1 Introduction

The q-Coloring problem asks whether the vertices of a graph can be properly colored using q colors. It is one of many colorability problems on graphs that have been widely studied. Since these are often NP-hard, they are good candidates to study from a parameterized perspective [2, 5]. Here we use additional parameters, other than the size of the input, to describe the complexity of the problem. In this paper we study preprocessing algorithms (called kernelizations or kernels) that aim to reduce the size of an input graph in polynomial time, without changing its colorability status.

The natural choice for a parameter for q-Coloring is the number of colors q. However, since even 3-Coloring is NP-hard, this parameter does not give interesting results. Therefore the problem is studied using different parameters, that often try to capture the complexity of the input graph. For example, Fiala et. al. [6] compared the parameterized

* This work was supported by NWO Veni grant "Frontiers in Parameterized Preprocessing" and NWO Gravitation grant "Networks".

12th International Symposium on Parameterized and Exact Computation (IPEC 2017).
Editors: Daniel Lokshtanov and Naomi Nishimura; Article No. 22; pp. 22:1–22:12
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

complexity of several coloring problems when parameterized by vertex cover, to the complexity when parameterized by treewidth. Jansen and Kratsch [8] studied graph coloring when parameterized by a hierarchy of different parameters.

In this earlier work [8], Jansen and Kratsch provided a kernel for q-Coloring parameterized by Vertex Cover with $\mathcal{O}(k^q)$ vertices that can be encoded in $\mathcal{O}(k^q)$ bits. Furthermore they showed that for $q \geq 4$, a kernel of bitsize $\mathcal{O}(k^{q-1-\varepsilon})$ is unlikely to exist. Unfortunately, these bounds left a gap of a factor k and it remained unclear whether the upper or the lower bound had to be strengthened. As our first main result, we close this gap. We show in Theorem 7 that the kernel for q-Coloring parameterized by Vertex Cover can be further improved to have $\mathcal{O}(k^{q-1})$ vertices and a bitsize of $\mathcal{O}(k^{q-1} \log k)$. This matches the previously known lower bound up to $k^{o(1)}$ factors.

To obtain this improvement, we use a recent result by the current authors [9] about the kernelization of constraint satisfaction problems when parameterized by the number of variables. A non-trivial data reduction can be achieved when the constraints are given by equalities of low-degree polynomials on boolean variables. The size of the resulting instance then depends on the maximum degree of the given polynomials. Suppose now we are given a 3-Coloring instance G with vertex cover S and let $I = V(G) \setminus S$ be the corresponding independent set. One can think of each vertex $v \in I$ as a constraint of the form "my neighbors use at most 2 different colors", such that a remaining color can be used to color v. We write these constraints as polynomial equalities and apply our previous result to find out which ones are redundant. Since vertices of the independent set can be colored independently, a vertex that corresponds to a redundant constraint can be removed from G, without changing the 3-colorability of G. To apply this idea to obtain a kernel for q-Coloring parameterized by Vertex Cover, the key technical step is to build a polynomial of degree $q-1$ that captures the desired constraint.

Our second main result concerns the parameterization by the number of vertices n. The current authors showed in earlier work [10] that for a number of graph problems it is impossible to give a kernel of size $\mathcal{O}(n^{2-\varepsilon})$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. This implies that the number of edges cannot efficiently be reduced to a subquadratic amount without changing the answer, a task that is also known as sparsification. For example, q-Coloring was shown to have no non-trivial sparsification for any $q \geq 4$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. The case for $q = 3$ remained open. One might think that 3-Coloring is so restrictive, that a 3-colorable instance is likely to either be sparse, or have a very specific structure. Exploiting this structure could then allow for a non-trivial sparsification. In Theorem 12 we show that this is not the case: 3-Coloring allows no kernel of size $\mathcal{O}(n^{2-\varepsilon})$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

From this bound it follows that the $\Omega(k^{q-1-\varepsilon})$ lower bound for the parameterization by vertex cover also holds for $q = 3$, since the size of a vertex cover is at most the total number of vertices in the graph. This completely settles the kernelization complexity of q-Coloring parameterized by Vertex Cover, up to $k^{o(1)}$ factors.

Related work.

Dell and Van Melkebeek showed that d-CNF-Satisfiability with n variables has no kernel of size $\mathcal{O}(n^{d-\varepsilon})$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [4]. Continuing this line of research, precise kernel lower bounds were shown for a variety of problems. For example, it was shown that Vertex Cover is unlikely to have a kernel of size $\mathcal{O}(k^{2-\varepsilon})$ [4], while a kernel with $\mathcal{O}(k^2)$ edges and $\mathcal{O}(k)$ vertices is known. Furthermore, the Point-Line cover problem, which asks to cover a set of n points in the plane with at most k lines, was proven to have a tight kernel lower bound of size $\mathcal{O}(k^{2-\varepsilon})$ [11], assuming $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$. Dell and Marx [3] proved polynomial kernelization lower bounds for several packing problems. They showed how a table structure can help realize the reduction that is needed for such a lower bound. We will also use this table structure in this paper.

2 Preliminaries

To denote the set of numbers 1 to n, we use the following notation: $[n] := \{i \in \mathbb{N} \mid 1 \leq i \leq n\}$. For $x, y \in \mathbb{Z}$ we write $x \equiv_2 y$ to denote that x and y are congruent modulo 2. For a finite set X and non-negative integer k, let $\binom{X}{k}$ be the collection of all subsets of X of size exactly k.

A graph G has vertex set $V(G)$ and edge set $E(G)$. All graphs considered in this paper are simple and undirected. For a vertex $u \in V(G)$, let $N_G(u) := \{v \in V(G) \mid \{u, v\} \in E(G)\}$ denote its open *neighborhood*. Let $G[S]$ for $S \subseteq V(G)$ denote the subgraph of G induced by S. A *vertex cover* of a graph G is a set $S \subseteq V(G)$ such that each edge has at least one endpoint in S (equivalently, $V(G) \setminus S$ is an *independent set* in G). A *proper q-coloring* of G is a function $c\colon V(G) \to [q]$ such that for all $\{u, v\} \in E(G)\colon c(u) \neq c(v)$.

A *parameterized problem* $\mathcal{Q}$ is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. Let $\mathcal{Q}, \mathcal{Q}' \subseteq \Sigma^* \times \mathbb{N}$ be parameterized problems and let $h\colon \mathbb{N} \to \mathbb{N}$ be a computable function. A *generalized kernel for* $\mathcal{Q}$ *into* $\mathcal{Q}$*' of size* $h(k)$ is an algorithm that, on input $(x, k) \in \Sigma^* \times \mathbb{N}$, takes time polynomial in $|x| + k$ and outputs an instance (x', k') such that:

1. $|x'|$ and k' are bounded by $h(k)$, and
2. $(x', k') \in \mathcal{Q}'$ if and only if $(x, k) \in \mathcal{Q}$.

The algorithm is a *kernel* for $\mathcal{Q}$ if $\mathcal{Q} = \mathcal{Q}'$. It is a *polynomial (generalized) kernel* if $h(k)$ is a polynomial. Since a polynomial-time reduction to an equivalent sparse instance yields a generalized kernel, a lower bound for the size of a generalized kernel can be used to prove the non-existence of sparsification algorithms.

We use the framework of cross-composition [1] to establish kernelization lower bounds, requiring the definitions of polynomial equivalence relations and OR-cross-compositions. We repeat them here for completeness:

▶ **Definition 1** (Polynomial equivalence relation, [1, Def. 3.1])**.** An equivalence relation $\mathcal{R}$ on Σ^* is called a *polynomial equivalence relation* if the following conditions hold.

- There is an algorithm that, given two strings $x, y \in \Sigma^*$, decides whether x and y belong to the same equivalence class in time polynomial in $|x| + |y|$.
- For any finite set $S \subseteq \Sigma^*$ the equivalence relation $\mathcal{R}$ partitions the elements of S into a number of classes that is polynomially bounded in the size of the largest element of S.

▶ **Definition 2** (Cross-composition, [1, Def. 3.3])**.** Let $L \subseteq \Sigma^*$ be a language, let $\mathcal{R}$ be a polynomial equivalence relation on Σ^*, let $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem, and let $f\colon \mathbb{N} \to \mathbb{N}$ be a function. An OR-*cross-composition of* L *into* $\mathcal{Q}$ (with respect to $\mathcal{R}$) *of cost* $f(t)$ is an algorithm that, given t instances $x_1, x_2, \ldots, x_t \in \Sigma^*$ of L belonging to the same equivalence class of $\mathcal{R}$, takes time polynomial in $\sum_{i=1}^{t} |x_i|$ and outputs an instance $(y, k) \in \Sigma^* \times \mathbb{N}$ such that:

- The parameter k is bounded by $\mathcal{O}(f(t) \cdot (\max_i |x_i|)^c)$, where c is some constant independent of t, and
- instance $(y, k) \in \mathcal{Q}$ if and only if there is an $i \in [t]$ such that $x_i \in L$.

▶ **Theorem 3** ([1, Theorem 6])**.** *Let $L \subseteq \Sigma^*$ be a language, let $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem, and let d, ε be positive reals. If L is NP-hard under Karp reductions, has an* OR*-cross-composition into $\mathcal{Q}$ with cost $f(t) = t^{1/d + o(1)}$, where t denotes the number of* *instances, and $\mathcal{Q}$ has a polynomial (generalized) kernelization with size bound $\mathcal{O}(k^{d-\varepsilon})$, then* NP $\subseteq$ coNP/poly.

We will refer to an OR-cross-composition of cost $f(t) = \sqrt{t}\log(t)$ as a *degree-2 cross-composition.* By Theorem 3, a degree-2 cross-composition can be used to rule out generalized kernels of size $\mathcal{O}(k^{2-\varepsilon})$.

3 Kernel for q-Coloring parameterized by Vertex Cover

In this section we develop a kernel for q-Coloring parameterized by Vertex Cover. The main tool is an earlier result [9] on constraint satisfaction problems (CSPs). In the right conditions, it can be used to reduce the number of constraints without changing the answer. We recall the required terminology. Define d-Polynomial Root CSP over the integers modulo 2 as the problem whose input consists of a set L of polynomial equalities over a set of boolean variables $V = \{x_1, \ldots, x_n\}$. Each equality is of the form $p(x_1, \ldots, x_n) \equiv_2 0$, where each polynomial has degree at most d. The question is whether all equalities can be satisfied by setting the input variables to 0 or 1. The following theorem follows directly from Theorem 2 together with Claim 3 in [9], where n is the total number of used variables.

▶ **Theorem 4.** *There is a polynomial-time algorithm that, given an instance (L, V) of d-*Polynomial root CSP *over an efficient field F, outputs $L' \subseteq L$ with at most $n^d + 1$ constraints such that any $0/1$-assignment to V satisfies L' if and only if it satisfies L.*

A field F is efficient if the field operations and Gaussian elimination can be done in polynomial time in the size of a reasonable input encoding. For our purposes it is only relevant that the integers modulo 2 form an efficient field.

To apply this machinery, we need to show how the coloring constraints expressed by an independent set of vertices can be encoded as polynomial equalities. To encode the color of a vertex v_i in this context we will use q boolean variables $y_{i,1}, \ldots, y_{i,q}$, one per possible color. The variable $y_{i,k}$ is set to *true* if vertex v_i has color k. We now define a choice assignment to the variables, to express that each vertex gets exactly one color.

▶ **Definition 5.** Let $\{y_{i,k} \mid i \in [n], k \in [q]\}$ be a set of boolean variables and let $\mathbf{y}$ be the vector containing all these variables. We say $\mathbf{y}$ is given a *choice assignment* if for all $i \in [n]$:

$$\sum_{k=1}^{q} y_{i,k} = 1.$$

Note that a choice assignment always sets exactly n variables to *true*. The following lemma gives a polynomial that can be used to express the constraint that out of exactly q neighbors of a given vertex u, there are at least two that have the same color. This constraint has to be satisfied to allow u to be properly q-colored. We will later apply such constraints to all possible subsets of q neighbors of u to obtain a safe reduction.

▶ **Lemma 6.** *Let $q > 0$ be an integer and let $y_{i,k}$ for $i \in [q], k \in [q]$ be boolean variables. Then there exists a polynomial p of degree $q - 1$ such that for any choice assignment to $\mathbf{y}$, we have $p(\mathbf{y}) \equiv_2 0$ if and only if there are $i, j, k \in [q]$ such that $y_{i,k} = y_{j,k} = 1$.*

Before proving Lemma 6, we give the polynomial p corresponding to $q = 3$ as an example.

$$p(\mathbf{y}) := \sum_{i_1 \neq i_2 \in [3]} \prod_{k=1}^{2} y_{i_k,k} = y_{1,1} \cdot y_{2,2} + y_{1,1} \cdot y_{3,2} + y_{2,1} \cdot y_{1,2} + y_{2,1} \cdot y_{3,2} + y_{3,1} \cdot y_{1,2} + y_{3,1} \cdot y_{2,2}.$$

Verify for this example that letting $y_{1,1} = y_{2,1} = y_{3,1} = 1$ and all other variables be zero, gives $p(\mathbf{y}) = 0 \equiv_2 0$. Setting $y_{1,1} = y_{2,2} = y_{3,2} = 1$ and all other variables to zero, gives $p(\mathbf{y}) = 2 \equiv_2 0$. Choosing $y_{1,1} = y_{2,2} = y_{3,3} = 1$ and all other variables zero, gives $p(\mathbf{y}) = 1 \equiv_2 1$, as desired. We now proceed with the general construction.

Proof of Lemma 6. Define the multivariate polynomial p as

$$p(\mathbf{y}) := \sum_{\substack{i_1,\ldots,i_{q-1} \in [q] \\ \text{distinct}}} \prod_{k=1}^{q-1} y_{i_k,k}.$$

To understand this polynomial and facilitate the remainder of the proof, it is useful to think of an associated set of variables $x_1, \ldots, x_q$ that take values from $[q]$ and represent a color. For each color variable x_i, the corresponding boolean variables $y_{i,1}, \ldots, y_{i,q}$ encode the value taken by x_i. In this notation, each monomial of p corresponds to a permutation of all but one of the color variables $x_1, \ldots, x_q$. The monomial evaluates to 1 if the i'th variable in this permutation has value i for all $i \in [q-1]$, and to 0 otherwise.

We proceed to show that p has the desired properties. It is easy to see the degree of p is $q-1$. It remains to prove the claim on the values of $p(\mathbf{y})$ for choice assignments. So consider a choice assignment to $\mathbf{y}$, and for each $i \in [q]$ let $x_i := k$ exactly when $y_{i,k} = 1$. This is well-defined as there is exactly one $k \in [q]$ such that $y_{i,k} = 1$. In these terms, we have to show that $p(\mathbf{y}) \equiv_2 0$ if and only if there are distinct color variables x_i, x_j such that $x_i = x_j$.

Suppose there do not exist $i, j \in [q]$ such that $x_i = x_j$, implying that $x_1, \ldots, x_q$ take q distinct values. For $k \in [q-1]$, let j_k be the unique index such that $x_{j_k} = k$, implying that $y_{j_k,k} = 1$. Then, $\prod_{k=1}^{q-1} y_{j_k,k} = 1$. For any other choice of distinct indices $i_1, \ldots, i_{q-1} \in [q]$, there exists $m \in [q-1]$ such that $i_m \neq j_m$. This implies that $y_{i_m,m} = 0$ and thereby $\prod_{k=1}^{q-1} y_{i_k,k} = 0$. Thus, $p(\mathbf{y}) = 1 \equiv_2 1$.

For the other direction, suppose there exist $i, j \in [q]$, such that $x_i = x_j$. We do a case distinction, where we consider the following cases: One color is used at least thrice, or there exist two colors that are both used more than once, or one color is used more than once and color q is used, or all colors except color q are used. More formally:

- There exist distinct $i, j, \ell \in [q]$ such that $x_i = x_j = x_\ell$. Then $p(\mathbf{y}) = 0$, because there do not exist distinct $i_1, \ldots, i_{q-1} \in [q]$ such that $x_{i_k} = k$ (and thus $y_{i_k,k} = 1$) for all $k \in [q-1]$. Hence all monomials of p evaluate to 0 and $p(\mathbf{y}) = 0$.
- There exist distinct $i, j, i', j' \in [q]$ such that $x_i = x_j$ and $x_{i'} = x_{j'}$. Then $p(\mathbf{y}) = 0$, because there do not exist distinct $i_1, \ldots, i_{q-1} \in [q]$ such that $x_{i_k} = k$ for all $k \in [q-1]$.
- There exist distinct $i, j \in [q]$ and there exists $\ell \in [q]$ such that $x_i = x_j$ and $x_\ell = q$. If $x_i = x_j = q$, it is not possible to find distinct $i_1, \ldots, i_{q-1} \in [q] \setminus \{i, j\}$ such that $x_{i_k} = k$ for all $k \in [q-1]$, thereby $p(\mathbf{y}) = 0$. If $x_i \neq q$, it is again not possible to find distinct $i_1, \ldots, i_{q-1} \in [q] \setminus \{\ell\}$ such that $x_{i_k} = k$ for all $k \in [q-1]$ since x_i and x_j are equal.
- Otherwise, there are distinct $i, j \in [q]$ and $k \in [q-1]$ such that $x_i = x_j = k$ and there is no $\ell \in [q]$ such that $x_\ell = q$. Furthermore, there are no distinct $i', j' \in [q] \setminus \{i, j\}$ such that $x_{i'} = x_{j'}$. In other words, each value from $[q-1]$ is assigned to exactly one color variable, except for the value k which occurs twice. For all $c \in [q-1]$ with $c \neq k$, let i_c be the unique index such that $x_{i_c} = c$ and thus $y_{i_c,c} = 1$. Then

$$y_{i,k} \cdot \prod_{\substack{c=1 \\ c \neq k}}^{q-1} y_{i_c,c} = y_{j,k} \cdot \prod_{\substack{c=1 \\ c \neq k}}^{q-1} y_{i_c,c} = 1.$$

However, $\prod_{c=1}^{q-1} y_{i_c,c} = 0$ for any other choice of $i_1, \ldots, i_{q-1}$. Thereby, $p(\mathbf{y}) = 2 \equiv_2 0$. ◀

We now give a kernel for the q-Coloring problem parameterized by the size of a vertex cover. The problem is defined as follows:

q-Coloring parameterized by Vertex Cover	**Parameter:** $\|S\|$
Input: A graph G with a vertex cover $S \subseteq V(G)$.	
Question: Does G have a proper q-coloring?	

We remark that in settings where no vertex cover of G is known, one can simply apply the kernelization using a 2-approximate vertex cover for S.

▶ **Theorem 7.** *For any constant $q \geq 3$, q-Coloring parameterized by the size of a vertex cover has a kernel with $\mathcal{O}(k^{q-1})$ vertices, which can be encoded in $\mathcal{O}(k^{q-1} \log k)$ bits. Furthermore, the resulting instance is a subgraph of the original input graph.*

Proof. Let input graph G with vertex cover S be given, where $|S| = k$. For each vertex $v \in S$, create boolean variables $C_{v,i}$ for $i \in [q]$. These variables can describe the color of v, by choosing $C_{v,i} = 1$ if v has color i and zero otherwise, which will give them a proper choice assignment. Let $\mathbf{C}$ contain all $q \cdot k$ constructed variables.

For each vertex $u \in V(G) \setminus S$, for each $X \in \binom{N_G(u)}{q}$, let $\mathbf{C_{u,X}}$ contain the variables constructed for set X in $N_G(u) \subseteq S$. Use Lemma 6 to obtain a polynomial $p_{u,X}$ of degree $q-1$, such that for any choice assignment to the variables we have $p_{u,X}(\mathbf{C_{u,X}}) \equiv_2 0$ if and only if there exist $i \in [q]$ and $v, w \in X$ such that $C_{v,i} = C_{w,i} = 1$.

Let L be the set of created polynomial equalities, thus $L := \{p_{u,X}(\mathbf{C_{u,X}}) \equiv_2 0 \mid u \in V(G) \setminus S \wedge X \in \binom{N_G(u)}{q}\}$. It is easy to see that L is an instance of $(q-1)$-Polynomial root CSP over the integers modulo 2. Use Theorem 4 in order to find $L' \subseteq L$ with $|L'| \leq (qk)^{q-1} + 1$, such that a boolean assignment to the variables in $\mathbf{C}$ satisfies L' if and only if it satisfies L. To obtain the kernel G', start with graph $G[S]$. For every equality $p_{u,X}(\mathbf{C_{u,X}}) \equiv_2 0 \in L'$, add u to G' if u is not yet present in G'. Furthermore, connect u to all vertices in X that u is not already adjacent to. It is easy to see that by this procedure, G' is a subgraph of G.

▶ Claim 8. G' is q-colorable if and only if G is q-colorable.

Proof. Since G' is a subgraph of G, graph G' is q-colorable if G is q-colorable.

For the opposite direction, let c' be a proper q-coloring of G'. For vertex $v \in S$ and color $i \in [q]$, define $C_{v,i} = 1$ if $c'(v) = i$ and $C_{v,i} = 0$ otherwise. By this definition, $\sum_{i=1}^{q} C_{v,i} = 1$ for all v, so all variable sets $\mathbf{C_{u,X}}$ are given a choice assignment. We will first show that this assignment satisfies all equalities in L'. Let $p_{u,X}(\mathbf{C_{u,X}}) \equiv_2 0 \in L'$. Then $u \in V(G') \setminus S$ and u is connected to all vertices in X in G'. Since u is colored by c', its neighbors do no have color $c'(u)$, thus $c'(u)$ is unused in the coloring of X. Since $|X| = q$ and we have exactly q colors, this implies that there exist $v, w \in X$ and color $i \in [q]$ such that $C_{v,i} = C_{w,i} = 1$. By Lemma 6, this implies $p_{u,X}(\mathbf{C_{u,X}}) \equiv_2 0$ as required.

From the choice of L' and Theorem 4 it now follows that all equalities in L are satisfied by this assignment. Let c denote the coloring c' restricted to the vertices in $G[S] = G'[S]$. We prove that c can be extended to a proper coloring of G. Since $V(G) \setminus S$ is an independent set, such an extension is possible if for each vertex $v \in V(G) \setminus S$ there exists a color that is not used on any vertex of $N_G(v)$.

Now assume for a contradiction that c cannot be extended to properly color some vertex u in $V(G) \setminus S$. Then for each color $i \in [q]$, there exists a vertex $v \in N_G(u)$ with $c(v) = i$ (or else we could use color i for u). Since $V(G) \setminus S$ is an independent set, $N_G(u) \subseteq S$. Pick a set $X \subseteq N_G(u)$ containing exactly one vertex of each color, thus $|X| = q$. By Lemma 6,

$p_{u,X}(\mathbf{C_{u,X}}) \equiv_2 1$ since there do not exist $v, w \in X$ and color $i \in [q]$ such that $C_{v,i} = C_{w,i} = 1$. But this contradicts the fact that all polynomial equalities in L are satisfied by the given assignment, since $p_{u,X}(\mathbf{C_{u,X}}) \equiv_2 0 \in L$. Hence c can be extended to properly color G. ⌟

▶ Claim 9. G' has at most $\mathcal{O}(k^{q-1})$ vertices and can be encoded in $\mathcal{O}(k^{q-1} \log k)$ bits.

Proof. Theorem 4 guarantees that $|L'| \leq (qk)^{q-1} + 1$ since there are qk boolean variables in total, and the polynomials have degree $q-1$. Thereby, $|V(G')| \leq k + (qk)^{q-1} + 1 = \mathcal{O}(k^{q-1})$, since q is a constant. Furthermore, $|E(G')| \leq |E(G'[S])| + q \cdot |L'| \leq k^2 + q \cdot ((qk)^{q-1} + 1) = \mathcal{O}(k^{q-1})$. An adjacency list encoding of the graph has size $\mathcal{O}(|E| \log |V| + |V|)$, which is $\mathcal{O}(k^{q-1} \cdot \log k^{q-1}) = \mathcal{O}(k^{q-1} \log k)$ for constant q. ⌟

It is easy to see that the kernel can be computed in polynomial time. Thereby, it follows from Claims 8 and 9 that we have given a kernel for q-Coloring of bitsize $\mathcal{O}(k^{q-1} \log k)$. ◀

4 Sparsification lower bound for 3-Coloring

In this section we provide a sparsification lower bound for 3-Coloring. We show that 3-Coloring does not have a (generalized) kernel of size $\mathcal{O}(n^{2-\varepsilon})$, unless NP ⊆ coNP/poly. This will also provide a kernel lower bound for 3-Coloring parameterized by vertex cover size, that matches the upper bound given in the previous section up to $k^{o(1)}$ factors.

For ease of presentation, we will prove the lower bound by giving a degree-2 cross-composition from a tailor-made problem to 3-List Coloring. The input to 3-List Coloring is a graph G together with a function L that assigns to each vertex v a list $L(v) \subseteq \{1, 2, 3\}$. The problem asks whether there exists a proper coloring of G, such that each vertex is assigned a color from its list. Before presenting the cross-composition, we introduce an important gadget that will be used. It was constructed by Jaffke and Jansen [7]. The gadget, which we will call a blocking-gadget, will be used to forbid one specific coloring of a given vertex set. The following Lemma is a rephrased version of Lemma 15 in [7].

▶ **Lemma 10.** *There is a polynomial-time algorithm that, given* $\mathbf{c} = (c_1, \ldots, c_m) \in [3]^m$*, outputs a* 3-List-Coloring *instance with* $\mathcal{O}(m)$ *vertices called blocking-gadget*($\mathbf{c}$) *that contains distinguished vertices* $(\pi_1, \ldots, \pi_m)$*. A coloring* $f : \{\pi_i \mid i \in [m]\} \to [3]$ *can be extended to a proper list coloring of blocking-gadget*($\mathbf{c}$) *if and only if* $f(\pi_i) = c_i$ *for some* $i \in [m]$*.*

The blocking-gadget can be used to forbid one specific coloring given by the tuple $\mathbf{c}$ of a set of vertices $v_1, \ldots, v_m$, by adding a blocking-gadget($\mathbf{c}$) and connecting π_i to v_i for all $i \in [m]$. If the color of v_i is c_i for all i, then the inserted edges prevent all π_i to receive the corresponding color c_i, and by Lemma 10 the coloring cannot be extended to the gadget. If however the color of v_i differs from c_i for some i, the gadget can be properly colored.

Having presented the gadget we use in our construction, we define the source problem for the cross-composition. This problem was also used as the starting problem for a cross-composition in our earlier sparsification lower bound for 4-Coloring [10].

2-3-Coloring with Triangle Split Decomposition [10]
Input: A graph G with a partition of its vertex set into $U \cup V$ such that $G[U]$ is an edgeless graph and $G[V]$ is a disjoint union of triangles.
Question: Is there a proper 3-coloring $c : V(G) \to \{1, 2, 3\}$ of G, such that $c(u) \in \{1, 2\}$ for all $u \in U$? We will refer to such a coloring as a *2-3-coloring* of the graph G, since two colors are used to color U, and three to color V.

▶ **Lemma 11** ([9, Lemma 3]). 2-3-Coloring with Triangle Split Decomposition *is NP-complete.*

To establish a quadratic lower bound on the size of generalized kernels, it suffices to give a degree-2 cross-composition from this special coloring problem into 3-Coloring. Effectively, we have to show that for any t, one can efficiently embed a series of t size-n instances indexed as $X_{i,j}$ for $i, j \in [\sqrt{t}]$, into a single 3-Coloring instance with $\mathcal{O}(\sqrt{t} \cdot n^{\mathcal{O}(1)})$ vertices that acts as the logical OR of the inputs. To achieve this composition, a common strategy is to construct vertex sets S_i and T_i of size $n^{\mathcal{O}(1)}$ for $i \in [\sqrt{t}]$, such that the graph induced by $S_i \cup T_j$ encodes input $X_{i,j}$. The fact that the inputs can be partitioned into an independent set and a collection of triangles facilitates this embedding; we represent the independent set within sets S_i and the triangles in sets T_i. To embed t inputs into a graph on $\mathcal{O}(\sqrt{t} \cdot n^{\mathcal{O}(1)})$ vertices, each vertex will have incident edges corresponding to many different input instances. The main issue when trying to find a cross-composition into 3-Coloring, is to ensure that when there is one 2-3-colorable input graph, the entire graph becomes 3-colorable. This is difficult, since the neighbors that a vertex in S_i has among the many different sets T_j should not invalidate the coloring. For vertices in some set T_j, we have a similar issue. Our choice of starting problem ensures that if some combination S_{i^*}, T_{j^*} corresponding to input X_{i^*,j^*} has a 2-3-coloring, then the remaining sets T_j can be safely colored 3, since vertices in S_{i^*} will use only two of the available colors. The key insight to ensure that vertices in the remaining S_i can also be colored, is to split them into multiple copies that each have at most one neighbor in any T_j. There will be at most one vertex in the neighborhood of a copy that is colored using color 1 or 2, thereby we can always color it using the other available color. Finally, additional gadgets will ensure that in some S_i all these copies get equal colors, and in some T_j the vertices that correspond to a triangle in the inputs are properly colored as such. With this intuition, we give the construction.

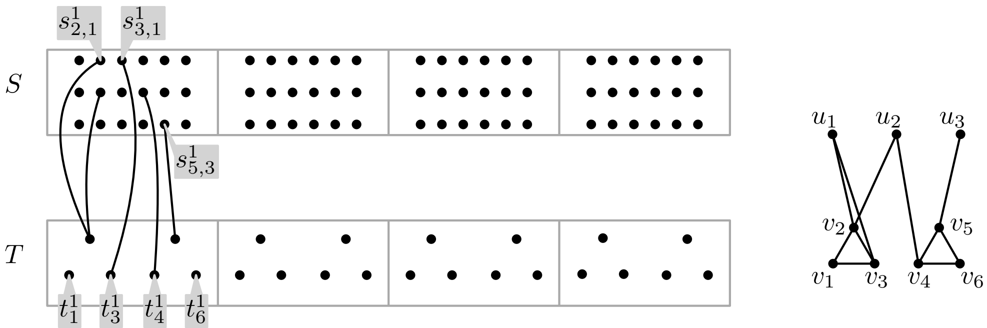
▶ **Theorem 12.** 3-Coloring *parameterized by the number of vertices n does not have a generalized kernel of size $\mathcal{O}(n^{2-\varepsilon})$ for any $\varepsilon > 0$, unless* NP $\subseteq$ coNP/poly.

Proof. To prove this statement, we give a degree-2 cross-composition from 2-3-Coloring with triangle split decomposition to 3-List Coloring and then show how to change this instance into a 3-Coloring instance. We start by defining a polynomial equivalence relation $\mathcal{R}$ on instances of 2-3-Coloring with triangle split decomposition. Let two instances be equivalent under $\mathcal{R}$, when the sets U have the same size and sets V consist of the same number of triangles. It is easy to verify that $\mathcal{R}$ is a polynomial equivalence relation.

By duplicating one of the inputs several times if needed, we ensure that the number of inputs to the cross-composition is a square. This increases the number of inputs by at most a factor four and does not change the value of the OR. Therefore, assume we are given t instances of 2-3-Coloring with Triangle Split Decomposition such that $t' := \sqrt{t}$ is integer. Enumerate these instances as $X_{i,j}$ for $i, j \in [t']$ and let instance $X_{i,j}$ have graph $G_{i,j}$. For input instance $X_{i,j}$, let U and V be such that U is an independent set with $|U| = m$ and V consists of n vertex-disjoint triangles. Enumerate the vertices in U as $u_1, \ldots, u_m$ and in V as $v_1, \ldots, v_{3n}$ such that $v_{3k-2}, v_{3k-1}, v_{3k}$ form a triangle for $k \in [n]$. We now create an instance of the 3-List Coloring problem, consisting of a graph G' together with a list function L that assigns a subset of the color palette $\{1, 2, 3\}$ to each vertex.

Refer to Figure 1 for a sketch of G'.

1. Initialize G' as the graph containing t' sets of $m \cdot 3n$ vertices each, called S_i for $i \in [t']$. Label the vertices in each of these sets as $s^i_{k,\ell}$ for $i \in [t']$, $k \in [3n]$ and $\ell \in [m]$. Define



(a) Constructed graph G' **(b)** Instance $X_{1,1}$

Figure 1 Construction of graph G' for $t' = 4$, $m = 3$, and $n = 2$. Edges between vertices in S and T are shown for instance $X_{1,1}$. All blocking-gadgets and the vertex sets A and B are left out.

$L(s^i_{k,\ell}) := \{1, 2\}$. The vertices $s^i_{1,\ell}, s^i_{2,\ell}, \ldots, s^i_{3n,\ell}$ together represent a single vertex of the independent set of an input instance, which is split into copies to ensure that every copy has at most one neighbor in each cell of T (the bottom row in Figure 1a).

2. Add t' sets of $3n$ vertices each, labeled T_j for $j \in [t']$. Label the vertices in T_j as t^j_k for $k \in [3n]$ and let $L(t^j_k) := \{1, 2, 3\}$. Vertices $t^j_{3k-2}, t^j_{3k-1}, t^j_{3k}$ correspond to a triangle in an input graph. They are not connected, so that we can safely color all vertices that do not correspond to a 3-colorable input with color 3.
3. Connect vertex $s^i_{k,\ell}$ to vertex t^j_k if in graph $G_{i,j}$ vertex u_ℓ is connected to v_k, for $k \in [3n]$ and $\ell \in [m]$. By this construction, the graph $G_{i,j}$ is isomorphic to the graph obtained from $G'[S_i \cup T_j]$ by replacing each triple $t^j_{3k-2}, t^j_{3k-1}, t^j_{3k}$ by a triangle for $k \in [n]$ and merging all $3n$ vertices $s^i_{k,\ell}$ in S_i that have the same value for $\ell \in [m]$.
4. Add vertex sets $A = \{a_1, \ldots, a_{t'}\}$ and $B := \{b_1, \ldots, b_{t'}\}$. These are used to choose indices i and j such that $G_{i,j}$ is 3-colorable. Let $L(a_i) := L(b_i) := \{1, 2\}$ for all $i \in [t']$.
5. Let $\mathbf{c}$ be defined by $c_i := 2$ for all $i \in [t']$. Add a blocking-gadget($\mathbf{c}$) to G'. Connect vertex a_i to the distinguished vertex π_i of this blocking-gadget for all $i \in [t']$.
6. Let $\mathbf{c}$ again be defined by $c_i := 2$ for all $i \in [t']$. Add a blocking-gadget($\mathbf{c}$) to G'. Connect vertex b_j to π_j for all $j \in [t']$. Together with the previous step, this ensures that in any proper list coloring at least one vertex in A and at least one vertex in B has color 1.
7. For every $i \in [t']$, $\ell \in [m]$, and $k \in [3n-1]$, for every $c_1, c_2 \in [2]$ with $c_1 \neq c_2$, add a blocking-gadget($(c_1, c_2, 1)$) to G'. Connect $s^i_{k,\ell}$ to π_1, $s^i_{k+1,\ell}$ to π_2, and a_i to π_3. This ensures that when a_i has color 1, vertices $s^i_{k,\ell}$ and $s^i_{k',\ell}$ have the same color for all $k, k' \in [3n]$.
8. For every $j \in [t']$, $k \in [n]$, for every $c_1, c_2, c_3 \in [3]$ that are not all pairwise distinct, add a blocking-gadget($(c_1, c_2, c_3, 1)$) to G'. Connect t^j_{3k-2} to π_1, t^j_{3k-1} to π_2, t^j_{3k} to π_3, and b_j to π_4. This construction ensures that if b_j is colored 1, all "triangles" in T_j are properly colored. If b_j is colored 2 however, the gadgets add no additional restrictions to the coloring of vertices in T_j.

This concludes the construction of G'; we proceed with the analysis.

▶ Claim 13. Let c be a proper 3-list coloring of G'. Then there exists $i \in [t']$ such that for all $\ell \in [m]$ and for all $k, k' \in [3n]$ we have $c(s^i_{k,\ell}) = c(s^i_{k',\ell})$.

Proof. By the blocking-gadget added in Step 5, there exists $i \in [t']$ such that $c(a_i) \neq 2$. Since $L(a_i) = \{1, 2\}$, this implies that $c(a_i) = 1$. We show that i has the required property.

Suppose there exist $k, k' \in [3n]$ and $\ell \in [m]$ such that $c(s^i_{k,\ell}) \neq c(s^i_{k',\ell})$. Then there must also exist $k \in [3n-1]$ such that $c(s^i_{k,\ell}) \neq c(s^i_{k+1,\ell})$, or else they would all be equal. Let (c_1, c_2, c_3) correspond to the coloring of $s^i_{k,\ell}$,$s^i_{k+1,\ell}$, and a_i as given by c. Then blocking-gadget$((c_1, c_2, c_3))$ was added in Step 7 and connected to these three vertices. But by Lemma 10, it follows that any list-coloring of this blocking-gadget must assign color c_i to some π_i for $i \in [3]$. By the way they are connected to $s^i_{k,\ell}$,$s^i_{k+1,\ell}$ and a_i, one edge has two endpoints of equal color, which is a contradiction. ⌟

We will say a triple of vertices v_1, v_2, v_3 is *colorful* (under coloring c), if they receive distinct colors, meaning $c(v_1) \neq c(v_2) \neq c(v_3) \neq c(v_1)$.

▶ Claim 14. Let c be a proper 3-list coloring of G'. Then there exists $j \in [t']$ such that for all $k \in [n]$ the triple t^j_{3k}, t^j_{3k-1}, t^j_{3k-2} is colorful.

Proof. By the blocking-gadget added in Step 6, there exists $j \in [t']$ such that $c(b_j) \neq 2$. Since $L(b_j) = \{1, 2\}$, this implies that $c(b_j) = 1$. We show that j has the desired property.

Suppose there exists $k \in [n]$, such that t^j_{3k}, t^j_{3k-1}, and t^j_{3k-2} are not a colorful triple. Let $(c_1, c_2, c_3, c_4) \in [3]^4$ correspond to the coloring given to t^j_{3k}, t^j_{3k-1}, t^j_{3k-1}, and b_j. In Step 8, blocking-gadget$((c_1, c_2, c_3, c_4))$ was added, together with connections to these four vertices. But by Lemma 10, any list-coloring of this blocking-gadget must assign color c_i to some π_i for $i \in [4]$. By the way they are connected to t^j_{3k}, t^j_{3k-1}, t^j_{3k-2}, and b_j, one edge has two endpoints of equal color, which is a contradiction. ⌟

▶ Claim 15. The graph G' is 3-list colorable $\Leftrightarrow$ some input instance $X_{i^* j^*}$ is 2-3-colorable.

Proof. ($\Rightarrow$) Suppose we are given a 3-list coloring c of G'. By Lemmas 13 and 14 there exist integers i^* and $j^* \in [t']$ such that for all $\ell \in [m]$ and for all $k, k' \in [3n]$ we have $c(s^{i^*}_{k,\ell}) = c(s^{i^*}_{k',\ell})$ and furthermore for all $k \in [n]$ the triple $t^{j^*}_{3k}$, $t^{j^*}_{3k-1}$, $t^{j^*}_{3k-2}$ is colorful. We show that this implies that G_{i^*,j^*} has a valid 2-3-coloring c', which we define as follows. Let $c'(u_\ell) := c(s^{i^*}_{1,\ell})$ for $\ell \in [m]$ and let $c'(v_k) := c(t^{j^*}_k)$ for $k \in [3n]$. It remains to verify that c' is a valid coloring of G_{i^*,j^*}. For any edge $\{u_\ell, v_k\} \in E(G_{i^*,j^*})$ with $\ell \in [m], k \in [3n]$, the endpoints receive different colors since

$$c'(v_k) = c(t^{j^*}_k) \neq c(s^{i^*}_{k,\ell}) = c(s^{i^*}_{1,\ell}) = c'(u_\ell).$$

For an edge $\{v_k, v'_k\} \in G_{i^*,j^*}$, its coloring corresponds to the coloring of $t^{j^*}_k$ and $t^{j^*}_{k'}$, which are colored differently by choice of j^* in Lemma 14. Furthermore, u_ℓ is always colored with color 1 or 2 as $L(s^{i^*}_{1,\ell}) = \{1, 2\}$. Thereby, c' is a proper 2-3-coloring of G_{i^*,j^*}.

($\Leftarrow$) Suppose c is a 2-3-coloring of G_{i^*,j^*}, such that the U-partite set of G_{i^*,j^*} is colored using only the colors 1 and 2. We will construct a 3-list coloring c' for graph G'. For $\ell \in [m]$ let $c'(s^{i^*}_{k,\ell}) := c(u_\ell)$ for all $k \in [3n]$. For $k \in [3n]$ let $c'(t^{j^*}_k) := c(v_k)$. For $j \neq j^*$ and $k \in [3n]$ let $c'(t^j_k) := 3$. For $i \neq i^* \in [t']$, $k \in [3n]$ and $\ell \in [m]$, pick $c'(s^i_{k,\ell}) \in \{1, 2\} \setminus \{c'(t^{j^*}_k)\}$. Let $c'(a_{i^*}) := 1$ and let $c'(b_{j^*}) := 1$. For $i \neq i^*$ let $c'(a_i) := 2$, similarly for $j \neq j^*$ let $c'(b_j) := 2$. Before coloring the vertices in blocking-gadgets, we will show that c' is proper on $G'[S \cup T]$. This will imply that the coloring defined so far is proper, as vertices in A and B only connect to blocking-gadgets.

Note that all edges in $G'[S \cup T]$ go from S to T. Consider an edge $\{s, t\}$ for $s \in S, t \in T$. Since $c'(s) \neq 3$, if $t \in T_j$ for $j \neq j^* \in [t']$, it follows immediately that $c'(s) \neq c'(t)$. Furthermore, if $s \in S_i$ for $i \neq i^* \in [t']$, $c'(s) \neq c'(t)$ by the definition of $c'(s)$. Otherwise, $s \in S_{i^*}$ and $t \in T_{j^*}$ and there exist $\{u, v\} \in E(G_{i^*,j^*})$ such that $c'(s) = c(u)$ and $c'(t) = c(v)$. Since c is a proper coloring, it follows that $c'(s) \neq c'(t)$.

To complete the proof, extend c' to also properly color all blocking-gadgets. This is possible for the blocking-gadgets added in Steps 5 and 6, since $c'(a_{i^*}) = 1$ and $c'(b_{j^*}) = 1$. Furthermore we show that this is possible for all blocking-gadgets introduced in Step 7. A blocking-gadget$((c_1, c_2, c_3))$ introduced in Step 7 either has π_3 connected to a_i for $i \neq i^*$ with $c'(a_i) = 2 \neq c_3$, or it is connected to a_{i^*} and in this case the vertices $s^{i^*}_{k,\ell}$ and $s^{i^*}_{k+1,\ell}$ are assigned equal colors and thus at least one of them has a coloring different from the coloring given by c_1 and c_2 as these colors are distinct. Thus, the colors that are forbidden on vertices π_i by the connections to the rest of the graph, do not correspond to (c_1, c_2, c_3) and c' can be extended to color the entire blocking-gadget by Lemma 10.

Similarly, coloring c' can be extended to blocking-gadgets$(\mathbf{c})$ added in Step 8, as either π_4 in the gadget is connected to b_j for $j \neq j^*$ and $c(b_j) = 2 \neq c_4$, or the three vertices from T connected to this gadget are colored with three different colors. ⌟

The claim above shows that we have given a cross-composition into 3-List Coloring. To obtain an instance of 3-Coloring, we add a triangle consisting of vertices $\{C_1, C_2, C_3\}$ to the graph. We connect a vertex v in G' to C_i if $i \notin L(v)$ for $i \in [3]$. This graph now has a proper 3-coloring if and only if the original graph had a proper 3-list coloring. Thus, by Claim 15, the resulting 3-Coloring instance acts as the logical or of the inputs.

It remains to bound the number of vertices of G'. In Step 1 we add $|S| = m \cdot 3n \cdot t'$ vertices and in Step 2 we add another $|T| = 3n \cdot t'$ vertices. Then in Step 4 we add $|A| + |B| = 2t'$ additional vertices. The two blocking-gadgets added in Steps 5 and 6 each have size $\mathcal{O}(t')$. The blocking-gadgets added in Step 7 have constant size, and we add six of them for each $i \in [t'], \ell \in [m], k \in [3n-1]$, thus adding $\mathcal{O}(t' \cdot m \cdot n)$ vertices. Similarly, the blocking-gadgets added in Step 8 have constant size, and we add a constant number of them for each $j \in [t'], \ell \in [n]$, thus adding $\mathcal{O}(t' \cdot n)$ vertices. This gives a total of $\mathcal{O}(t' \cdot n \cdot m) = \mathcal{O}(\sqrt{t} \cdot (\max_{i,j} |X_{i,j}|)^{\mathcal{O}(1)})$ vertices. Theorem 12 now follows from Theorem 3 and Lemma 11. ◀

The set of all vertices of a graph is always a valid vertex cover for that graph. Thereby, it follows from Theorem 12 that the lower bound also holds when parameterized by vertex cover. In [8, Theorem 3], it was shown that for any $q \geq 4$, q-Coloring parameterized by vertex cover does not have a generalized kernel of size $\mathcal{O}(k^{q-1-\varepsilon})$, unless NP ⊆ coNP/poly. Combining these results gives a lower bound for q-Coloring that matches the kernel size presented in the first section.

▶ **Corollary 16.** *For any $q \geq 3$, q-Coloring parameterized by vertex cover does not have a generalized kernel of bitsize $\mathcal{O}(k^{q-1-\varepsilon})$ for any $\varepsilon > 0$, unless* NP ⊆ coNP/poly.

5 Conclusion

We have given a kernel for q-Coloring parameterized by Vertex Cover with $\mathcal{O}(k^{q-1})$ vertices and bitsize $\mathcal{O}(k^{q-1} \log k)$, improving on the previously known kernel by almost a factor k. Furthermore, 3-Coloring when parameterized by the number of vertices has no kernel of size $\mathcal{O}(n^{2-\varepsilon})$, unless NP ⊆ coNP/poly. It was already known that for $q \geq 4$, q-Coloring parameterized by Vertex Cover was unlikely to yield a kernel of size $\mathcal{O}(k^{q-1-\varepsilon})$. Combining these results allows us to give the same lower bound for $q = 3$, under the assumption that NP ⊈ coNP/poly. Thereby we have provided an upper and lower bound on the kernel size of q-Coloring parameterized by Vertex Cover for any $q \geq 3$, that match up to $k^{o(1)}$ factors.

It is easy to see that the kernel lower bounds also hold for q-List Coloring, where every vertex v in the graph has a list $L(v) \subseteq [q]$ of allowed colors. Furthermore, we can also apply our kernel, by first reducing an instance of q-List Coloring to an instance of q-Coloring using q additional vertices, and adding these q vertices to the vertex cover of the graph. This only changes the size of the obtained kernel by a constant factor.

In this paper we gave a first example where applying the known results for sparsification of CSPs gives an improved kernel for a graph problem. It would be interesting to see if this technique can be applied to obtain smaller kernels for other graph problems as well. To apply this idea, one needs to first identify which constraints should be modeled. When the constraints are found, they need to be written as equalities of low-degree polynomials over a suitably chosen field. This requires the clever construction of polynomials that have a sufficiently low degree, in order to obtain a good bound on the kernel size.

Another direction for future research consists of obtaining optimal kernel bounds for q-Coloring with different structural parameters. For example, one could look at q-Coloring parameterized by a modulator to a cograph. This parameterization admits a polynomial kernel [8, Corollary 3], but tight bounds are not known.

References

1 Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.*, 28(1):277–305, 2014. doi:10.1137/120880240.

2 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.

3 Holger Dell and Dániel Marx. Kernelization of packing problems. In *Proc. 23th SODA*, SODA '12, pages 68–81, 2012.

4 Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014. doi:10.1145/2629620.

5 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

6 Jiří Fiala, Petr A. Golovach, and Jan Kratochvíl. Parameterized complexity of coloring problems: Treewidth versus vertex cover. *Theoretical Computer Science*, 412(23):2513 – 2523, 2011. doi:10.1016/j.tcs.2010.10.043.

7 Lars Jaffke and Bart M. P. Jansen. Fine-grained parameterized complexity analysis of graph coloring problems. In *Proc. 10th CIAC*, pages 345–356, 2017. doi:10.1007/978-3-319-57586-5_29.

8 Bart M. P. Jansen and Stefan Kratsch. Data reduction for graph coloring problems. *Inf. Comput.*, 231:70–88, 2013. doi:10.1016/j.ic.2013.08.005.

9 Bart M. P. Jansen and Astrid Pieterse. Optimal sparsification for some binary CSPs using low-degree polynomials. In *Proc. 41st MFCS*, pages 71:1–71:14, 2016. doi:10.4230/LIPIcs.MFCS.2016.71.

10 Bart M. P. Jansen and Astrid Pieterse. Sparsification upper and lower bounds for graph problems and not-all-equal SAT. *Algorithmica*, 79(1):3–28, 2017. doi:10.1007/s00453-016-0189-9.

11 Stefan Kratsch, Geevarghese Philip, and Saurabh Ray. Point line cover: The easy kernel is essentially tight. *ACM Trans. Algorithms*, 12(3):40:1–40:16, 2016. doi:10.1145/2832912.