



Optimal Data Reduction for Graph Coloring Using Low-Degree Polynomials

Bart M. P. Jansen¹ · Astrid Pieterse¹

Received: 23 December 2017 / Accepted: 17 April 2019 / Published online: 3 May 2019
© The Author(s) 2019

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 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 $k^{\mathcal{O}(1)}$ factors assuming $\text{NP} \not\subseteq \text{coNP/poly}$, and improves on the previous-best kernel of size $\mathcal{O}(k^q)$. We generalize this result for deciding q -colorability of a graph G , to deciding the existence of a homomorphism from G to an arbitrary fixed graph H . Furthermore, we can replace the parameter vertex cover by the less restrictive parameter twin-cover. We prove that H -COLORING PARAMETERIZED BY TWIN-COVER has a kernel of size $\mathcal{O}(k^{\Delta(H)} \log k)$. Our second result shows that 3-COLORING does not admit non-trivial sparsification: assuming $\text{NP} \not\subseteq \text{coNP/poly}$, 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.

Keywords Graph coloring · Graph homomorphism · Kernelization

Mathematics Subject Classification 05C15 · 05C85

This work was supported by NWO Veni grant “Frontiers in Parameterized Preprocessing” and NWO Gravitation grant “Networks”. An extended abstract appeared in the Proceedings of the 12th International Symposium on Parameterized and Exact Computation, IPEC 2017.

✉ Bart M. P. Jansen
b.m.p.jansen@tue.nl

Astrid Pieterse
astridpieterse@outlook.com

¹ Eindhoven University of Technology, P. O. Box 513, 5600 MB Eindhoven, The Netherlands

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 [4,7]. 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. [8] compared the parameterized complexity of several coloring problems when parameterized by vertex cover, to the complexity when parameterized by treewidth. Jansen and Kratsch [13] studied graph coloring when parameterized by a hierarchy of different parameters.

In this earlier work [13], 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 by improving the kernel.

To obtain this improvement, we can use a recent result by the current authors [14] 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. Amongst other results, it was shown that constraints of the q -NOT-ALL-EQUAL SAT problem, which consist of at most q literals that should not all evaluate to the same truth value, can be written as equalities of polynomials of degree $q - 1$. This allows the number of clauses in an n -variable instance of q -NOT-ALL-EQUAL SAT to be efficiently reduced to $\mathcal{O}(n^{q-1})$ [14, Corollary 6]. In earlier work [13, Theorem 3], it was shown that kernelization lower bounds for q -NOT-ALL-EQUAL SAT parameterized by the number of variables imply lower bounds for q -COLORING PARAMETERIZED BY VERTEX COVER. In this paper, we effectively show that the relation between these problems also goes in the other way, and use ideas from the new upper-bound technology for q -NOT-ALL-EQUAL SAT to derive improved kernelizations for q -COLORING.

To see how graph coloring relates to constraint satisfaction, suppose 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 . We can 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 constraints.

In this paper, we further generalize the problem by studying the H -COLORING problem. The problem asks for a given graph G and fixed graph H , whether there exists a homomorphism $f: V(G) \rightarrow V(H)$ such that $\{u, v\} \in E(G) \Rightarrow \{f(u), f(v)\} \in E(H)$. Instead of using the size of a vertex cover as the parameter, we use a smaller parameter called twin-cover [9]. We show in Theorem 2 that H -COLORING parameterized by the size of a twin-cover has a kernel with $\mathcal{O}(k^{\Delta(H)})$ vertices and bitsize $\mathcal{O}(k^{\Delta(H)} \log k)$, where $\Delta(H)$ is the maximum degree of graph H . Since q -COLORING is equivalent to K_q -COLORING where K_q is the clique on q vertices, this result immediately gives a kernel of bitsize $\mathcal{O}(k^{q-1} \log k)$ for q -COLORING parameterized by vertex cover. This closes the gap with the lower bound for q -COLORING up to $k^{o(1)}$ factors.

Often, when describing a kernel for a problem parameterized by a structural parameter like vertex cover, it is assumed that (an approximation of) the minimum vertex cover is given with the input [2, 11]. However, an interesting feature of our kernel for H -COLORING is that it can be computed without knowing an (approximation of the) optimal twin-cover of the input graph. The fact that the graph has a size- k twin-cover is only used to analyze the size of the resulting kernel.

Our second main result concerns the parameterization by the number of vertices n . The current authors showed in earlier work [15] that for a number of graph problems it is impossible to give a kernel of size $\mathcal{O}(n^{2-\varepsilon})$, unless $\text{NP} \subseteq \text{coNP/poly}$. 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 $\text{NP} \subseteq \text{coNP/poly}$. 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 3 we show that this is not the case: 3-COLORING allows no kernel of size $\mathcal{O}(n^{2-\varepsilon})$, unless $\text{NP} \subseteq \text{coNP/poly}$.

Related work Hell and Nešetřil showed that H -COLORING is NP-hard for any non-bipartite graph H that has no self-loops [10]. For a bipartite graph, the problem is equivalent to testing whether the input graph is bipartite, and thus polynomial-time solvable. Chitnis et al. show that the problem of finding a smallest set $W \subseteq V(G)$ such that $G - W$ is H -list-colorable is FPT when H is (C_6, P_6) -free and bipartite, when parameterized by the size of H together with the solution size [3].

Ganian introduced TWIN-COVER as a new parameter [9] and gave relations to existing parameters. For example, a minimum twin-cover is not larger than a minimum vertex cover, but twin-cover is incomparable to treewidth. The paper also gives an FPT algorithm for PRECOLORING EXTENSION parameterized by the size of a twin-cover, and studies a number of other problems using this parameter.

Dell and Van Melkebeek showed that for any $d \geq 3$, d -CNF-SATISFIABILITY with n variables has no kernel of size $\mathcal{O}(n^{d-\varepsilon})$, unless $\text{NP} \subseteq \text{coNP/poly}$ [6]. 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 bitsize $\mathcal{O}(k^{2-\varepsilon})$ [6], 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 $\mathcal{O}(k^{2-\varepsilon})$ [16], assuming $\text{NP} \not\subseteq \text{coNP/poly}$. Dell and Marx [5] 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 our lower bound.

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 and let $\binom{X}{\leq k}$ be the collection of all subsets of X of size at most k .

2.1 Graphs

All graphs considered in this paper are finite, simple, and undirected. In particular, this means that graphs do not have self-loops. A graph G has vertex set $V(G)$ and edge set $E(G)$. For disjoint sets $X, Y \subseteq V(G)$, let $E_G(X, Y) := \{\{x, y\} \in E(G) \mid x \in X, y \in Y\}$ denote the edges with one endpoint in X and one endpoint in Y . Let $G[S]$ for $S \subseteq V(G)$ denote the subgraph of G induced by S . For vertex set $X \subseteq V(G)$ we use $G - X := G[V(G) \setminus X]$ to denote the result of removing the vertices in X from G . For $F \subseteq E(G)$, let $G \setminus F$ denote the result of removing all edges in F from G . Let $\Delta(G)$ denote the maximum degree of any vertex in G and let $\omega(G)$ denote the size of a largest clique in G .

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* and let $N_G[u] := N_G(u) \cup \{u\}$ denote its *closed neighborhood*. For a vertex set $S \subseteq V(G)$, let $N_G(S) := \{v \in V(G) \setminus S \mid \{u, v\} \in E(G)\}$ denote its open neighborhood.

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, $G - S$ is an *independent set*). We say vertices u and $v \in V(G)$ are (*true*) *twins* whenever $N_G[u] = N_G[v]$. Note that this relation is transitive. We say $X \subseteq V(G)$ is a *twin-cover* [9] of G , if for every edge $\{u, v\} \in E(G)$, vertex $u \in X$, or $v \in X$, or u and v are twins.

A q -*coloring* of G is a function $f: V(G) \rightarrow [q]$ such that for all $\{u, v\} \in E(G)$ we have $f(u) \neq f(v)$. Let G and H be graphs. We say that G is H -*colorable* if there exists a function $f: V(G) \rightarrow V(H)$ such that for all $\{u, v\} \in E(G)$ it holds that $\{f(u), f(v)\} \in E(H)$. Such a function is also called a *homomorphism* from G to H . Note that G has a homomorphism to K_q (a clique on k vertices) if and only if G is q -colorable. We will regularly refer to the image of a vertex $v \in V(G)$ under a homomorphism from G to H as the *color* of this vertex. In this paper, we will only consider H -COLORING where H has no self-loops and is not bipartite, as otherwise the

problem is polynomial-time solvable. We will frequently use the following properties of H -colorings in the remainder of the paper.

Observation 1 *Let $S \subseteq V(G)$ such that $G[S]$ is a clique and let f be an H -coloring of G . Define $X := \{f(v) \mid v \in S\}$. Then $H[X]$ is a clique in H and all vertices in S receive a different color, so that $|S| = |X|$.*

Observation 2 *Let $v \in V(G)$ and let f be an H -coloring of G . Then the number of colors used to color $N_G(v)$ is bounded by $\Delta(H)$.*

2.2 Parameterized Complexity

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 $h: \mathbb{N} \rightarrow \mathbb{N}$ 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 $f: \mathbb{N} \rightarrow \mathbb{N}$ 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 $x_1, x_2, \dots, x_t \in \Sigma^*$ 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 1 [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 $\text{NP} \subseteq \text{coNP/poly}$.*

We will refer to an OR-cross-composition of cost $f(t) = \sqrt{t} \log(t)$ as a *degree-2 cross-composition*. By Theorem 1, a degree-2 cross-composition can be used to rule out generalized kernels of size $\mathcal{O}(k^{2-\varepsilon})$.

3 Kernel for H -Coloring Parameterized by Twin-Cover

In this section, we give a kernel for H -COLORING parameterized by the size of a twin-cover. We start by showing how to partition the graph into vertex sets that are twins in Sect. 3.1. We introduce some of the polynomial equalities that we use and their properties in Sect. 3.2, and use them in Sect. 3.3 to define the set of equalities that is constructed for a given input graph. In Sect. 3.4 we define the three reduction rules our kernel will use and prove that they are safe. Finally, in Sect. 3.5 we give the kernel.

3.1 Twin Decomposition

Computing a minimum TWIN-COVER is NP-hard, since VERTEX COVER is NP-hard on graphs where no two vertices are twins. We will therefore construct the kernel for H -COLORING without knowing a twin-cover of the input graph. In order to do this, we decompose the graph into vertex sets consisting of twins. Recall that throughout the paper, twins are vertices with the same *closed* neighborhood.

Definition 3 (*(Partial) twin decomposition*) *A partial twin decomposition of a graph G is a partition $\Pi = \{P_1, \dots, P_m\}$ of $V(G)$, such that any two vertices in the same partite set are twins. Partition Π is a *twin decomposition* if furthermore any two vertices in different partite sets are not twins.*

To be able to use the twin decomposition for the kernelization procedure, we recall the folklore fact that it can be computed efficiently.

Lemma 1 *A twin decomposition of a graph G can be computed in $\mathcal{O}(|V(G)| + |E(G)|)$ time.*

Proof (sketch) This is for example stated in [17, Exercise 2.17] for the case of finding false twins, which are vertices such that $N_G(u) = N_G(v)$. Finding (true) twins is similar. An example solution uses the adjacency-list representation, and adds each vertex to its own adjacency list. Then we efficiently sort the adjacency lists by bucket

sort. We partition these sorted lists into sets of duplicates by a recursive process, which splits the sets into groups based on the first element of each adjacency list, which is removed from the recursively partitioned lists. The running time is linear since the sum of the lengths of all adjacency lists is $\mathcal{O}(|E(G)|)$, while the work done in each iteration is proportional to the decrease in total volume of the lists. \square

The next lemma shows how the twin decomposition and a minimal twin-cover may intersect.

Lemma 2 *Let G be a graph with twin decomposition Π and a minimal twin-cover S . Then for any partite set $P \in \Pi$ it holds that either $P \subseteq S$ or $P \cap S = \emptyset$.*

Proof Let $P \in \Pi$. Suppose $P \cap S \neq \emptyset$ and $P \setminus S \neq \emptyset$. Let $u \in S \cap P$ and $v \in P \setminus S$. We show that $S \setminus \{u\}$ is a twin-cover of G , which contradicts the assumption that S is minimal.

Let $\{u, w\}$ be any edge in G . If $w = v$, then since $u, v \in P$, this is an edge between twins. If $w \neq v$, then since u and v are twins, it follows that $\{v, w\} \in E(G)$. Thereby, either $w \in S$ and thus edge $\{u, w\}$ is covered by w , or w and v are twins. In this case, by transitivity of being twins, u and w are also twins. This proves that $S \setminus \{u\}$ is indeed a twin-cover of G , which is a contradiction. \square

3.2 Modeling Constraints as Polynomial Equalities

As explained in the introduction, the kernelization is based on a connection to constraint satisfaction problems. To find the kernel, we represent the constraints that a vertex set puts on the coloring of its neighborhood, as polynomial equalities. We then use this representation to find redundant vertices and edges in the graph. To use this idea, we need some additional lemmas and definitions. Recall that a monomial of degree d is the product of d variables, with the unique monomial of degree zero being the constant 1. For example, $x_1 \cdot x_3 \cdot x_3$ is a monomial of degree three. A monomial is *multilinear* if each variable occurs at most once. We start by proving a bound on the number of multilinear monomials of degree at most d over a set of n variables.

Lemma 3 (cf. [14, Claim 4]) *There are at most $n^d + 1$ multilinear monomials of degree at most d over a set of n variables.*

Proof The number of multilinear monomials over n variables of degree at most d is equal to $\sum_{i=0}^d \binom{n}{i}$. We will show that $\sum_{i=1}^d \binom{n}{i} \leq n^d$. The left side counts all non-empty subsets of $[n]$ of size at most d . Each of these can be mapped to a distinct d -tuple containing numbers from $[n]$, by repeating an arbitrary element. Since there are n^d possible d -tuples, the claim follows. \square

To find redundant vertices and edges, we will model the constraints that a vertex puts on the coloring of its neighborhood by equalities of low-degree polynomials. It then remains to find redundant equalities. To be able to do this, we need the following additional definitions.

Definition 4 (*Span*) Let X be a set of vectors in $\{0, 1\}^d$ for some $d \in \mathbb{N}$. Define $\text{span}_2(X)$ as the set of all vectors $\mathbf{y} \in \{0, 1\}^d$ for which there exist $\mathbf{x}_1, \dots, \mathbf{x}_\ell \in X$ such that $\mathbf{y} \equiv_2 \sum_{i=1}^{\ell} \mathbf{x}_i$, i.e., \mathbf{y} is a linear combination of $\mathbf{x}_1, \dots, \mathbf{x}_\ell$ when vectors are added component-wise, over the integers modulo 2.

Let $p(x_1, \dots, x_n)$ be a multivariate polynomial in (a subset of) the variables x_1, \dots, x_n , evaluated over the integers modulo 2, of degree at most d for some fixed d . Hence p is a weighted sum of monomials of degree at most d over x_1, \dots, x_n . For some fixed ordering of the monomials of degree d over x_1, \dots, x_n , let $\text{vect}(p)$ denote the vector containing the coefficients of the corresponding monomials in p . We may use this notation as well for polynomial equalities where the right-hand side is zero; for a polynomial equality $p(x_1, \dots, x_n) \equiv_2 0$ over the integers modulo 2, such that p has degree at most d , we let $\text{vect}(p(x_1, \dots, x_n) \equiv_2 0)$ be defined as $\text{vect}(p)$.

If P is a set of multivariate polynomials in variables x_1, \dots, x_n , we use $\text{vect}(P)$ to denote $\{\text{vect}(p) \mid p \in P\}$. Similarly, if L is a set of polynomial equalities in variables x_1, \dots, x_n for which the right-hand sides are zero, we let $\text{vect}(L)$ be defined as $\{\text{vect}(p) \mid (p(x_1, \dots, x_n) \equiv_2 0) \in L\}$.

The following lemma follows from the definition above.

Lemma 4 Let P be a set of polynomials of degree at most d over a tuple of variables \mathbf{y} , and let q be a polynomial of degree at most d over \mathbf{y} . If $\text{vect}(q) \in \text{span}_2(\text{vect}(P))$, then any assignment to \mathbf{y} that satisfies $p(\mathbf{y}) \equiv_2 0$ for all $p \in P$, satisfies $q(\mathbf{y}) \equiv_2 0$.

Proof Choose $\alpha_p \in \{0, 1\}$ for all $p \in P$ such that $\text{vect}(q) \equiv_2 \sum_{p \in P} \alpha_p \text{vect}(p)$. Consider an assignment to the variables \mathbf{y} with $p(\mathbf{y}) \equiv_2 0$ for all $p \in P$. Let \mathbf{y}' be the vector containing the evaluation of the monomials of degree at most d over \mathbf{y} , for the values assigned to \mathbf{y} . List them in the same order in which the coefficients for these monomials are listed in $\text{vect}(\cdot)$. Since a polynomial is a weighted sum of monomials, the value of a polynomial p of degree most d in \mathbf{y} for the assigned values, equals the inner product of $\text{vect}(p)$ and \mathbf{y}' . So:

$$q(\mathbf{y}) = \text{vect}(q) \cdot \mathbf{y}' \equiv_2 \sum_{p \in P} \alpha_p \text{vect}(p) \cdot \mathbf{y}' \equiv_2 \sum_{p \in P} \alpha_p \cdot p(\mathbf{y}) \equiv_2 0. \quad \square$$

To utilize polynomials over boolean variables to represent solutions of graph H -coloring problems, we represent the color of a vertex v in a graph G by $|V(H)|$ boolean variables, indicating whether v has the corresponding color. We now define a partial choice assignment, which reflects that any vertex receives at most one color.

Definition 5 (*Partial choice assignment*) Let $y_{i,k} \in \{0, 1\}$ for $i \in [n], k \in [q]$ be a set of boolean variables and let \mathbf{y} be the vector containing all these variables. We say \mathbf{y} is given a *partial choice assignment* if for all $i \in [n]$:

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

Note that a partial choice assignment sets at most n variables to *true*. By this definition, a partial choice assignment can be seen as a partial coloring in the following way: $y_{i,k} = 1$ means vertex i has color k . Note that the coloring of some vertices may remain undefined.

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. By combining multiple such constraints, we can ensure that at most $q - 1$ different colors are used in the neighborhood of vertex u , leaving one color free for u itself in the q -coloring problem. When evaluating the polynomial for \mathbf{y} that is given a partial choice assignment, the polynomial has the following two essential properties. (1) It equals 1 modulo 2 when the q vertices all receive a distinct color, and (2) it equals 0 modulo 2 whenever two vertices have the same color, or when two vertices have no color defined.

Lemma 5 *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 $q - 1$ over the integers modulo 2, such that whenever the variables in \mathbf{y} are given a partial choice assignment, it holds that $p(\mathbf{y}) \equiv_2 1$ if and only if*

- there exist no $i, j, k \in [q]$ with $i \neq j$ such that $y_{i,k} = y_{j,k} = 1$, and
- for all $k \in [q - 1]$ there exists $i \in [q]$ such that $y_{i,k} = 1$.

Before proving Lemma 5, we give the polynomial p corresponding to $q = 3$ as an example.

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

One may verify for this example that letting $y_{1,1} = y_{2,2} = y_{3,3} = 1$ and all other variables be zero, gives $p(\mathbf{y}) = 1 \equiv_2 1$, as desired. Setting $y_{1,1} = y_{2,2} = y_{3,2} = 1$ and all other variables to zero, gives $p(\mathbf{y}) = 2 \equiv_2 0$, which explains why the modulus is used. Letting $y_{1,1} = y_{2,3} = y_{3,3} = 1$ and all other variables zero, results in $p(\mathbf{y}) = 0 \equiv_2 0$. Furthermore, letting $y_{1,1} = y_{2,2} = 1$ and all other variables be zero, also results in $p(\mathbf{y}) \equiv_2 1$. We now proceed with the general construction.

Proof of Lemma 5 Define the multivariate polynomial p as

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

We prove that p has the desired properties. It is easy to see that the degree of p is $q - 1$. It remains to prove the claim on the values of $p(\mathbf{y})$ for partial choice assignments. So let \mathbf{y} be given a partial choice assignment, and for each $i \in [q]$ let $x_i := k$ exactly when $y_{i,k} = 1$. Let $x_i := 0$ when there is no such $y_{i,k}$.

We now show that $p(\mathbf{y}) \equiv_2 1$ if there exist no $i, j, k \in [q]$ with $i \neq j$ such that $y_{i,k} = y_{j,k} = 1$, and for all $k \in [q - 1]$ there exists $i \in [q]$ such that $y_{i,k} = 1$. In terms of the values for x_i , this implies that they are all distinct, and that $[q - 1] \subseteq \{x_1, \dots, x_q\}$. Thereby, we have $\{x_1, \dots, x_q\} = [q]$ or $\{x_1, \dots, x_q\} = [q - 1] \cup \{0\}$.

For $k \in [q - 1]$, let j_k be the unique index such that $x_{j_k} = k$, implying that $y_{j_k,k} = 1$. Note that this is well defined, since all values from $[q - 1]$ are used exactly once. Then, $\prod_{k=1}^{q-1} y_{j_k,k} = 1$. For any other choice of distinct indices $i_1, \dots, i_{q-1} \in [q]$, 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, $p(\mathbf{y}) = 1 \equiv_2 1$.

Now suppose there exist $i, j \in [q]$, such that $x_i = x_j \neq 0$, or there exists $k \in [q - 1]$ such that $y_{i,k} = 0$ for all $i \in [q]$. We show that $p(\mathbf{y}) \equiv_2 0$ by a case distinction.

- Suppose there exists $k \in [q - 1]$ such that $\sum_{i=1}^q y_{i,k} = 0$, or equivalently there is no $i \in [q]$ such that $x_i = k$. Thereby, for any choice of $i_1, \dots, i_{q-1} \in [q]$, we have that $\prod_{\ell=1}^{q-1} y_{i_\ell,\ell} = 0$, since $y_{i_k,k} = 0$. Thus, $p(\mathbf{y}) \equiv_2 0$.
- Suppose there exists no $k \in [q - 1]$ such that $\sum_{i=1}^q y_{i,k} = 0$. Thereby, for each $k \in [q - 1]$ there exists $i \in [q]$ such that $x_i = k$. It follows from our earlier assumption that there must exist $i, j, k \in [q]$ with $i \neq j$ such that $x_i = x_j = k$, which implies $k < q$. 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, \dots, i_{q-1} . Thereby, $p(\mathbf{y}) = 2 \equiv_2 0$. □

There is another way to look at the proof of Lemma 5, which may give some intuition. Consider the $q \times q$ matrix A given by $a_{i,j} = y_{i,j}$ for all $i \in [q], j \in [q - 1]$, and $a_{i,q} = 1$ for all $i \in [q]$. Then $p(\mathbf{y})$ is equal to the determinant of the matrix A over the integers modulo 2. It follows from this that $p(\mathbf{y}) \equiv_2 1$ if and only if the columns of A are linearly independent. One may verify that if \mathbf{y} is given a partial choice assignment, the columns of A are linearly independent if and only if the two conditions of the lemma statement hold for \mathbf{y} .

3.3 Construction of Polynomial Equalities

We continue to define the polynomial equalities that will be constructed for a subset P of the vertices of G . These are necessary constraints on the coloring of $N_G(P)$, such this coloring can be extended to H -color P . In this construction, P will be a partite set of the twin decomposition of G , and hence a clique.

Let G be a graph with $P \subseteq V(G)$. We create variables $c_{v,i}$ for each $v \in V(G)$ and $i \in V(H)$, denoting whether vertex v has color i . Let \mathbf{C} be the set containing all constructed variables. Let $L(P, G)$ be the set of polynomial equalities produced

by the following procedure, which results in two types of constraints. The first type will ensure that the neighborhood of P uses at most $\Delta(H)$ colors. The second type of constraints will ensure that the coloring of the neighborhood of P can indeed be extended to color P : we ensure that there is a clique in H of size at least $|P|$, such that the colors used for $N_G(P)$ are in the neighborhood of this clique in H . The reason for having these two types of constraints is to obtain a good bound on the degree of the relevant polynomials.

Consider each set $S \subseteq N_G(P)$ with $|S| = \Delta(H) + 1$ and each set $X \subseteq V(H)$ with $|X| = |S|$. Pick an arbitrary ordering on X such that $X = \{x_1, \dots, x_{|S|}\}$ and use Lemma 5 to find a polynomial $p_{P,S,X}$ such that $p_{P,S,X}(\mathbf{C}) \equiv_2 1$ if and only if the following two statements hold:

1. there exist no $u \neq v \in S, k \in X$ such that $c_{u,k} = c_{v,k} = 1$, and
2. for all $k \in [|S| - 1]$ there exists $u \in S$ such that $c_{u,x_k} = 1$.

When considering a partial choice assignment that corresponds to a mapping from $V(G)$ to $V(H)$, the above two statements together imply that S is colored with $|S| = \Delta(H) + 1$ distinct colors (which is what we want to avoid). More precisely, when $p_{P,S,X}(\mathbf{C}) \equiv_2 1$, the coloring of S would use colors $x_1, \dots, x_{|S|-1}$ and one other color. Add the following constraint to $L(P, G)$:

$$p_{P,S,X}(\mathbf{C}) \equiv_2 0.$$

For the second type of constraints, consider each set $S \subseteq N_G(P)$ of $\ell \in [\Delta(H)]$ elements. Pick an arbitrary ordering of S , such that $S = \{s_1, \dots, s_\ell\}$. Consider each sequence $\mathbf{x} = (x_1, \dots, x_\ell) \in V(H)^\ell$ (of not necessarily distinct elements). Let $Y := \bigcap_{i=1}^\ell N_H(x_i)$ be the common neighborhood of all vertices from \mathbf{x} in H . If $H[Y]$ does not contain a clique of size at least $|P|$ (i.e. if $\omega(H[Y]) < |P|$), add the following polynomial equality to $L(P, G)$:

$$q_{P,S,\mathbf{x}}(\mathbf{C}) := \prod_{i=1}^\ell c_{s_i,x_i} \equiv_2 0.$$

The computation of $\omega(H[Y])$ for some $Y \subseteq V(H)$ can be done in constant time, since H is considered constant. This concludes the construction of $L(P, G)$. Note that the constraints $L(P, G)$ are defined solely in terms of the variables that describe the coloring of the *open neighborhood* of P .

Next, we define a complete list of equations for G .

Definition 6 ($L_\Pi(G)$) Let G be a graph and let Π be a partition of its vertex set. Let $L_\Pi(G)$ be defined as follows.

$$L_\Pi(G) := \bigcup_{P \in \Pi} L(P, G).$$

Since the polynomials for the first type of constraints have degree at most $|S| - 1 = \Delta(H)$ by Lemma 5, while the polynomials for the second type of constraints are the product of $\ell \leq \Delta(H)$ variables, we observe the following.

Observation 3 Let G be a graph with Π a partition of its vertex set. The polynomials in $L_\Pi(G)$ have degree at most $\Delta(H)$.

We now prove two useful properties for this choice of constraints.

Lemma 6 Let G be a graph with some partial twin decomposition Π . Let $f : V(G) \rightarrow V(H)$ be a mapping. If f is an H -coloring of G , then the boolean assignment to $\mathbf{C} := \{c_{v,i} \mid v \in V(G), i \in V(H)\}$ given by $c_{v,i} = 1 \Leftrightarrow f(v) = i$ satisfies all constraints in $L_\Pi(G)$.

Proof Let f be given and the value of any $c_{v,i} \in \mathbf{C}$ be defined by $c_{v,i} = 1 \Leftrightarrow f(v) = i$. We show that this assignment satisfies all constraints in $L_\Pi(G)$, by showing that it satisfies both types of constraints in $L(P, G)$ for all $P \in \Pi$. Consider some $P \in \Pi$. Since it consists of twins, it is a clique in G . As H has no self-loops, the vertices in P all receive distinct colors by Observation 1, and the colors used on P form a clique in H . The fact that P consists of twins also implies that $\{u, v\} \in E(G)$ for all $u \in P, v \in N_G(P)$. Thereby, any color used in P is not used in the coloring of $N_G(P)$.

Consider a constraint $p_{P,S,X}(\mathbf{C}) \equiv_2 0 \in L(P, G)$ for $S \subseteq N_G(P)$ of size $|\Delta(H)| + 1$ and $X \subseteq V(H)$ of the same size. By Observation 2, the vertices in S use at most $\Delta(H) = |S| - 1$ colors. Hence at least one color $d \in V(H)$ appears twice on a vertex of S . If $d \in X$ then some color of X is used twice on S , violating the first condition of Lemma 5. If $d \notin X$ then at least two vertices $u, v \in S$ do not receive a color from X and hence $\sum_{i \in X} c_{u,i} = \sum_{i \in X} c_{v,i} = 0$. Since $|S| = |X| = q$, there are at most $q - 2$ vertices $w \in S$ for which there exists $i \in X$ with $c_{w,i} = 1$. As such, there exist distinct colors $d_1, d_2 \in X$ such that $\sum_{s \in S} c_{s,d_1} = \sum_{s \in S} c_{s,d_2} = 0$, so that the lowest-indexed of d_1 and d_2 violates the second condition of Lemma 5. It then follows from Lemma 5 that $p_{P,S,X}(\mathbf{C}) \equiv_2 0$ as required.

Consider a constraint $q_{P,S,X}(\mathbf{C}) \equiv_2 0 \in L(P, G)$ for $S \subseteq N(P)$ and $\mathbf{x} = (x_1, \dots, x_{|S|}) \in V(H)^{|S|}$. Suppose this constraint is not satisfied. Then the coloring of S is given by \mathbf{x} and furthermore, $H[Y]$ where $Y := \bigcap_{i=1}^{|S|} N_H(x_i)$ does not contain a clique on $|P|$ vertices. But any H -coloring (for H without self-loops) colors any clique in G with an equally-sized clique in H , and the colors used on the clique P must be adjacent in H to all the colors used on the neighbors S of P in G . Since $H[Y]$ contains no clique on $|P|$ vertices, f cannot be an H -coloring of G . It follows that for any H -coloring, all constraints are satisfied. \square

Let $S \subseteq V(G)$ and let $f : S \rightarrow V(H)$ be an H -coloring of $G[S]$. We say that f can be extended to color G , if there exists $f' : V(G) \rightarrow V(H)$ such that f' is an H -coloring of G and furthermore $f'(v) = f(v)$ for all $v \in S$. The next lemma shows that an H -coloring of a part of the graph can be extended to color the entire graph, if it satisfies certain constraints.

Lemma 7 Let G be a graph with $P' \subseteq V(G)$. Let $f : V(G) \setminus P' \rightarrow V(H)$ be an H -coloring of $G - P'$, such that the boolean assignment to $\mathbf{C} := \{c_{v,i} \mid v \in V(G), i \in V(H)\}$ given by $c_{v,i} = 1 \Leftrightarrow (v \notin P' \wedge f(v) = i)$ satisfies all constraints in $L(P', G)$. Then f can be extended to H -color G .

Proof Let f be given and \mathbf{C} be defined by $c_{v,i} = 1 \Leftrightarrow (v \notin P' \wedge f(v) = i)$. We start by showing that $N_G(P')$ uses at most $\Delta(H)$ different colors. Suppose not, then there is a set $S \subseteq N_G(P')$ of size $\Delta(H) + 1$ using $\Delta(H) + 1$ distinct colors. Let X be the set of colors used in S . It follows from Lemma 5, that regardless of which ordering of X was chosen when constructing $p_{P',S,X}(\mathbf{C})$, we have $p_{P',S,X}(\mathbf{C}) \equiv_2 1$. By definition, $L(P', G)$ contains the equation $p_{P',S,X}(\mathbf{C}) \equiv_2 0$. This contradicts the assumption that all constraints in $L(P', G)$ are satisfied.

Let X be the set of colors used by $N_G(P')$, suppose $|X| = \ell$. We have shown above that $\ell \leq \Delta(H)$. Let S be a size- ℓ subset of $N_G(P')$ such that for every color $x \in X$, there exists exactly one $s \in S$ such that $f(s) = x$. Let $Y := \bigcap_{x \in X} N_H(x)$. Suppose towards a contradiction that $H[Y]$ contains no clique of size $|P'|$. As such, for some ordering of S as $S = \{s_1, \dots, s_\ell\}$ and for $\mathbf{x} = (x_1, \dots, x_\ell)$ such that $f(s_i) = x_i$ for all i , the constraint $q_{P',S,\mathbf{x}}(\mathbf{C}) \equiv_2 0$ was added to $L(P', G)$. However, by definition, $q_{P',S,\mathbf{x}}(\mathbf{C}) \equiv_2 1$, contradicting the fact that all constraints in $L(P', G)$ are satisfied. Thereby, $H[Y]$ contains a clique K of size $|P'|$, where $Y := \bigcap_{i=1}^{|X|} N_H(x_i)$. To extend f to color P' , assign each vertex in P' a distinct color from K .

It remains to verify that f is indeed a valid H -coloring of G . Any edge between two vertices in $V(G) \setminus P'$ remains properly colored. Any edge in P' is properly colored, because its endpoints have a different color and K is a clique in H . Any edge between P' and $V(G) \setminus P'$ is properly colored, because all vertices in K are common neighbors of the vertices in X , and $K \cap X = \emptyset$. □

3.4 Reduction Rules

We now present the three reduction rules that will be used to obtain the kernel, and prove that they are safe. The first checks whether the graph is trivially not H -colorable, the second removes sets of edges from the graph, and the third removes sets of vertices from the graph.

Reduction rule 1 *Let G be a graph with twin decomposition Π . If there exists $P \in \Pi$ with $|P| > \omega(H)$, return a trivial no-instance.*

It is easy to see that Reduction rule 1 preserves the answer to the problem, since in this case G cannot have an H -coloring by Observation 1.

Reduction rule 2 *Let G be a graph with twin decomposition Π . Let $P' \neq P'' \in \Pi$ such that $E_G(P', P'') \neq \emptyset$. If*

$$\text{vect}(L(P', G)) \subseteq \text{span}_2 \left(\text{vect}(L_\Pi(G \setminus E_G(P', P''))) \right),$$

remove all edges in $E_G(P', P'')$ from graph G .

Reduction rule 2 is the key rule for our kernelization. It simplifies the graph by removing all edges between two distinct sets of twins P' and P'' , if the constraints $L(P', G)$ are implied by the constraints generated by the remaining graph $G \setminus E_G(P', P'')$. Observe that it is essential for the effectiveness of Reduction rule 2 that Π is a twin decomposition, since applying the rule to partial twin

decompositions that are not twin decompositions may increase the size of an optimal twin-cover in the considered graph. The following lemma proves that the reduction rule is safe.

Lemma 8 *If G' is obtained from G by applying Reduction rule 2, then G is H -colorable if and only if G' is H -colorable.*

Proof Let G' be $G \setminus E_G(P', P'')$. Clearly, if G is H -colorable, then G' is also H -colorable, since G' is a subgraph of G .

In the other direction, let f' be an H -coloring of G' . It follows from Lemma 6 and the fact that Π is a partial twin decomposition of $G \setminus E_G(P', P'')$ that the derived setting of the boolean variables \mathbf{C} satisfies the constraints in $L_\Pi(G \setminus E_G(P', P''))$. Since $\text{vect}(L(P', G)) \subseteq \text{span}_2(\text{vect}(L_\Pi(G \setminus E_G(P', P''))))$ it follows from Lemma 4 that this setting of \mathbf{C} also satisfies all constraints in $L(P', G)$. Let f be defined as f' restricted to the vertices in $G' - P'$. Note that $G' - P'$ equals $G - P'$ by definition. It is easy to see that f is indeed an H -coloring of $G - P'$ since $G - P'$ is a subgraph of G' and f' is an H -coloring of G' . Furthermore, f satisfies the constraints in $L(P', G)$ since it colors the relevant vertices the same as f' . It now follows from Lemma 7 that we can extend f to color all vertices in G . Thereby, G is H -colorable. \square

The final rule effectively removes isolated cliques from the graph, when H has a sufficiently large clique to allow them to be colored properly.

Reduction rule 3 *Let G be a graph with twin decomposition Π . If there exists $P' \in \Pi$ with $N_G(P') = \emptyset$ and $|P'| \leq \omega(H)$, then remove P' from G .*

Lemma 9 *If G' is obtained from G by applying Reduction rule 3, then G is H -colorable if and only if G' is H -colorable.*

Proof Let P' be such that $G' = G - P'$. Clearly, if G is H -colorable, G' remains H -colorable. In the other direction, suppose G' is H -colorable. We show how to extend this coloring to G . Since we assumed that $|P'| \leq \omega(H)$, graph H has a clique X of size $|P'|$. Use the colors of X to assign a different color to each vertex in P' . Since $N_G(P') = \emptyset$, this gives an H -coloring of G . \square

Lemma 10 *Let G' be the graph resulting from applying Reduction rule 2 or 3 to a graph G . Then the size of a minimum twin-cover in G' is at most the size of a minimum twin-cover in G .*

Proof Let $P \subseteq V(G)$. It is easy to see that if S is a twin-cover of G , then it is also a twin-cover of $G' = G - P$. Thereby, the statement holds for Reduction rule 3.

Let Π be the twin decomposition of G and let $P' \neq P'' \in \Pi$. Let $F := E_G(P', P'')$ be the set of edges between P' and P'' . Let S be a twin-cover of G . We show that S is a twin-cover of $G' = G \setminus F$. Clearly, any edges that were previously covered, are still covered. We show that all vertices that were twins in G , are also twins in $G \setminus F$ to conclude the proof. Let u, v be twins in G , and let $P \in \Pi$ such that $u, v \in P$. If $P \neq P'$ and $P \neq P''$, it is obvious that u and v remain twins in $G \setminus F$. Suppose u, v lie in P' or in P'' ; without loss of generality, suppose $u, v \in P'$. Note that the edge $\{u, v\}$ belongs to $E(G) \setminus F$. Then $N_{G \setminus F}[u] = N_G[u] \setminus P''$. Since u and v are twins in G , we have $N_G[u] \setminus P'' = N_G[v] \setminus P'' = N_{G \setminus F}[v]$. Thereby, u and v are twins in $G \setminus F$. It follows that the lemma statement also holds for Reduction rule 2. \square

3.5 Analysis of the Kernelization

Theorem 2 *For any fixed non-bipartite graph H (without self-loops), H -COLORING parameterized by the size k of a twin-cover has a kernel with $\mathcal{O}(k^{\Delta(H)})$ vertices and edges, which can be encoded in $\mathcal{O}(k^{\Delta(H)} \log k)$ bits. Furthermore, the kernelized instance is a subgraph of the original input graph.*

Proof Let G be a graph. Apply Reduction rules 1, 2, and 3 exhaustively. Let the resulting graph be G' . We show that this is a correct kernelization. \square

Claim 1 *Reduction rules 1–3 can exhaustively be applied in time $|V(G)|^{\mathcal{O}(\Delta(H))}$.*

Proof We can compute a twin decomposition of G in linear time by Lemma 1. Computing $\omega(H)$ can be done in $\mathcal{O}(1)$ time for fixed H . Hence Reduction rule 1 can be applied in polynomial time.

The set $L_{\Pi}(G)$ contains at most $m := 2|V(G)| \cdot |V(G)|^{\Delta(H)+1} \cdot |V(H)|^{\Delta(H)+1}$ polynomial equalities (the number of ways to pick S, X , and P as for the definition of $p_{P,S,X}$ and $q_{P,S,X}$), over $|V(G)| \cdot |V(H)|$ variables. All polynomials we employ are multilinear. This can be verified directly from their construction, and explained by noting that squaring a number does not change it, when working modulo 2. By Lemma 3, we therefore only have to consider $(|V(G)| \cdot |V(H)|)^{\Delta(H)} + 1$ coefficients for the polynomials. Constructing the required polynomial equalities can be done in time $|V(G)|^{\mathcal{O}(\Delta(H))}$ for fixed H . We can test if one vector lies in the span of a set of other vectors by comparing the ranks of matrices of dimensions at most $m \times ((|V(G)| \cdot |V(H)|)^{\Delta(H)} + 1)$. Thereby, Reduction rule 2 can be applied in polynomial time. Note that the twin decomposition has to be recomputed after each application of Rule 2. Reduction rule 3 can trivially be applied in polynomial time. Since $|\Pi| \leq |V(G)|$, checking for all $P \in \Pi$ whether any of the reduction rules can be applied takes polynomial time.

Each rule can be applied at most $|V(G)|^2$ times, as it always removes at least one edge or vertex. The claim follows. \square

Let G' be the result of applying Reduction rules 1, 2, and 3 exhaustively. We use the following claim to prove a bound on the size of G' .

Claim 2 *The resulting graph G' has $\mathcal{O}(|V(H)|^{\Delta(H)} \cdot \Delta(H)^2 \cdot k^{\Delta(H)})$ vertices and $\mathcal{O}(|V(H)|^{\Delta(H)} \cdot \Delta(H)^3 \cdot k^{\Delta(H)})$ edges.*

Proof When Reduction rule 1 has been applied at any point, G' trivially has constant size. Otherwise, since G has a twin-cover of size k , it follows from Lemma 10 that G' has a twin-cover of size at most k . Let Y be a minimum twin-cover of G' , such that $|Y| \leq k$. Let Π be the twin decomposition of G' . By Lemma 2, every P in Π is either fully contained in Y , or disjoint from Y . Let $\Pi' := \{P \in \Pi \mid P \cap Y = \emptyset\}$. Define $L_{tc} := \bigcup_{P \in \Pi'} L(P, G')$, and note that $N_{G'}(P) \subseteq Y$ for all $P \in \Pi'$. This implies the polynomial equalities in L_{tc} only involve the variables controlling the coloring of Y . By Observation 3, the polynomials in L_{tc} have degree at most $\Delta(H)$ and they use at most $|V(H)|$ variables for each of the k vertices in Y . Define

$$\alpha := (k \cdot |V(H)|)^{\Delta(H)} + 1.$$

Let $V_{Ic} := \text{vect}(L_{Ic})$ be the vectors of coefficients corresponding to the polynomials in L_{Ic} . Compute a basis $V'_{Ic} \subseteq V_{Ic}$ of V_{Ic} over the integers modulo 2. Let $L'_{Ic} \subseteq L_{Ic}$ contain all polynomial equalities whose corresponding vector is contained in V'_{Ic} . Since all employed polynomials are multilinear, it follows that the vectors in V_{Ic} only have nonzero entries in positions corresponding to multilinear monomials over $|Y| \cdot |V(H)|$ distinct variables, of which there are at most α by Lemma 3. As the size of the basis V'_{Ic} equals the rank of the matrix containing the (row)vectors V_{Ic} , which is upper-bounded by the number of columns that contain a nonzero entry, it follows that $|L'_{Ic}| = |V'_{Ic}| \leq \alpha$.

We define a set of meta-edges $F \subseteq (\Pi' \times (\Pi \setminus \Pi'))$ based on the constraints in L'_{Ic} . For each constraint Z in L'_{Ic} , do the following.

- Suppose $Z = (p_{P',S,X}(\mathbf{C}) \equiv_2 0)$ for some $P' \in \Pi'$, $S \subseteq N_G(P')$, and $X \subseteq V(H)$. Since P' is a partite set of twins that is disjoint from Y , we have $N_G(P') \subseteq Y$ since Y is a twin-cover. So each $v \in S$ belongs to a partite set P_v of twins with $P_v \in \Pi \setminus \Pi'$. For each $v \in S$, add (P', P_v) to F .
- Otherwise, $Z = (q_{P',S,\mathbf{x}}(\mathbf{C}) \equiv_2 0)$ for some $P' \in \Pi'$, $S \subseteq N_G(P')$, and sequence $\mathbf{x} = (x_1, \dots, x_{|S|}) \in V(H)^{|S|}$. Similarly as above, for each $v \in S$ take $P_v \in \Pi \setminus \Pi'$ such that $v \in P_v$ and add (P', P_v) to F .

The above procedure adds at most $\Delta(H) + 1$ meta-edges for each constraint in L'_{Ic} . Thereby,

$$|F| \leq \alpha(\Delta(H) + 1). \tag{1}$$

We now argue that for any $(P', P'') \notin F$ with $P' \in \Pi'$ and $P'' \in \Pi \setminus \Pi'$, the following holds:

$$L'_{Ic} \subseteq L_{\Pi}(G' \setminus E_{G'}(P', P'')). \tag{2}$$

To see this, consider a constraint in L'_{Ic} . It is of one of two possible types, and it was added to $L_{Ic} = \bigcup_{P \in \Pi'} L(P, G') \supseteq L'_{Ic}$ because it satisfied the criteria described in Sect. 3.3. Effectively, the constraint was created because some set $P \in \Pi'$ contains a certain vertex set S of size at most $\Delta(H) + 1$ in its open neighborhood in G' . But by our choice of meta-edges F , the set P still has S in its neighborhood in $G' \setminus E_{G'}(P', P'')$, so that all constraints of L'_{Ic} are also contained in $L_{\Pi}(G' \setminus E_{G'}(P', P''))$.

Using this, we show that for all $P' \in \Pi'$ and $P'' \in \Pi \setminus \Pi'$:

$$E_{G'}(P', P'') \neq \emptyset \Rightarrow (P', P'') \in F. \tag{3}$$

Suppose there exist $P' \in \Pi'$, $P'' \in \Pi \setminus \Pi'$ such that $E_{G'}(P', P'') \neq \emptyset$ but $(P', P'') \notin F$. It follows from Eq. 2 that $L'_{Ic} \subseteq L_{\Pi}(G' \setminus E_{G'}(P', P''))$. Thereby,

$$\begin{aligned} \text{span}_2(\text{vect}(L_{\Pi}(G' \setminus E_{G'}(P', P'')))) &\supseteq \text{span}_2(\text{vect}(L'_{Ic})) \\ &= \text{span}_2(V'_{Ic}) \supseteq V_{Ic} = \text{vect}(L_{Ic}) \supseteq \text{vect}(L(P', G')). \end{aligned}$$

Thereby, Reduction rule 2 could be applied to G' , which is a contradiction. It follows that $P' \in \Pi'$ and $P'' \in \Pi \setminus \Pi'$ can only be connected in G' if there is a corresponding meta-edge in F . We can now use Eqs. 1 and 3 to bound the number of vertices and edges in G' .

First of all, for all $P' \in \Pi'$ there must exist some $P'' \in \Pi \setminus \Pi'$ such that $(P', P'') \in F$, otherwise it follows from Eq. 3 that $N_{G'}(P') = \emptyset$ and P' would have been removed by Reduction rule 3. Thereby $|\Pi'| \leq |F|$. Since $|P| \leq \omega(H) \leq \Delta(H) + 1$ for all $P \in \Pi$ by Reduction rule 1, the number of vertices of G' can be bounded as follows.

$$\begin{aligned} |V(G')| &\leq |F| \cdot (\Delta(H) + 1) + |Y| \leq \left((k|V(H)|)^{\Delta(H)} + 1 \right) \cdot (\Delta(H) + 1)^2 + k \\ &= \mathcal{O} \left(|V(H)|^{\Delta(H)} \cdot \Delta(H)^2 \cdot k^{\Delta(H)} \right). \end{aligned}$$

If edge $\{u, v\} \in G'$ with $u \in Y$ and $v \notin Y$, then there exist $(P', P'') \in F$ such that $u \in P', v \in P''$. Since $|P| \leq \Delta(H) + 1$ for any $P \in \Pi$, there are at most $|F| \cdot (\Delta(H) + 1)^2$ such edges. Furthermore, there are at most $\binom{|Y|}{2} \leq k^2$ edges between vertices in Y , and at most $|F| \cdot (\Delta(H) + 1)^2$ edges between vertices in $V(G) \setminus Y$. Thereby, the total number of edges can be bounded by:

$$\begin{aligned} |E(G')| &\leq |F| \cdot (\Delta(H) + 1)^2 + |Y|^2 + |F| \cdot (\Delta(H) + 1)^2 \\ &\leq 2\alpha(\Delta(H) + 1)^3 + k^2 \\ &= 2 \left((k \cdot |V(H)|)^{\Delta(H)} + 1 \right) \cdot (\Delta(H) + 1)^3 + k^2 \end{aligned}$$

(note that $\Delta(H) \geq 2$ for non-bipartite H)

$$= \mathcal{O} \left(|V(H)|^{\Delta(H)} \cdot \Delta(H)^3 \cdot k^{\Delta(H)} \right).$$

This concludes the proof of Claim 2. □

It follows from the correctness of Reduction rules 1, 2, and 3 that G' is H -colorable if and only if G is H -colorable. It follows from Claims 1 and 2 that we have given a kernel for H -coloring with $\mathcal{O}(k^{\Delta(H)})$ vertices and edges for constant-size H that can be computed in polynomial time. By encoding the graph using adjacency lists, it can be encoded in $\mathcal{O}(k^{\Delta(H)} \cdot \log k)$ bits. □

The following corollary shows that Theorem 2 generalizes the result obtained for q -COLORING parameterized by vertex cover in the extended abstract of this work.

Corollary 1 *For any constant $q \geq 3$, q -COLORING parameterized by the size of a twin-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 Since q -COLORING is equivalent to K_q -COLORING, and $\Delta(K_q) = q - 1$ and K_q has q vertices, the result now follows directly from Theorem 2. □

4 Sparsification Lower Bound for 3-Coloring

In this section we provide a sparsification lower bound for 3-COLORING. We show that 3-COLORING parameterized by the number of vertices n does not have a (generalized) kernel of size $\mathcal{O}(n^{2-\varepsilon})$, unless $\text{NP} \subseteq \text{coNP/poly}$. This will also provide a kernel lower bound for 3-COLORING parameterized by the size of a twin-cover, 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, and then show how to modify the resulting 3-LIST COLORING instance to a 3-COLORING instance. 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 is to decide whether there exists a 3-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 [12]. 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 14 in [12].¹

Lemma 11 *There is a polynomial-time algorithm that, given $\mathbf{c} = (c_1, \dots, c_m) \in [3]^m$, outputs a 3-LIST-COLORING instance with $\mathcal{O}(m)$ vertices called $\text{blocking-gadget}(\mathbf{c})$ that contains distinguished vertices (π_1, \dots, π_m) . A coloring $f: \{\pi_i \mid i \in [m]\} \rightarrow [3]$ can be extended to a list coloring of $\text{blocking-gadget}(\mathbf{c})$ 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, \dots, v_m , by adding a $\text{blocking-gadget}(\mathbf{c})$ 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 11 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 [15].

2-3-COLORING WITH TRIANGLE SPLIT DECOMPOSITION [15]

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 3-coloring $c: V(G) \rightarrow \{1, 2, 3\}$ 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 12 ([15, Lemma 3]) *2-3-COLORING WITH TRIANGLE SPLIT DECOMPOSITION is NP-complete.*

¹ For a proof of the lemma, refer to Lemma 15 in the full version of the cited paper: <https://arxiv.org/abs/1701.06985>.

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_j 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_j . The triangles will be represented by an independent set connected to a number of gadgets, not triangles, to ensure that the sets T_j can be uni-colored, unless index i is “selected”.

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 3 *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, \dots, u_m and in V as v_1, \dots, v_{3n} such that $v_{3k-2}, v_{3k-1}, v_{3k}$

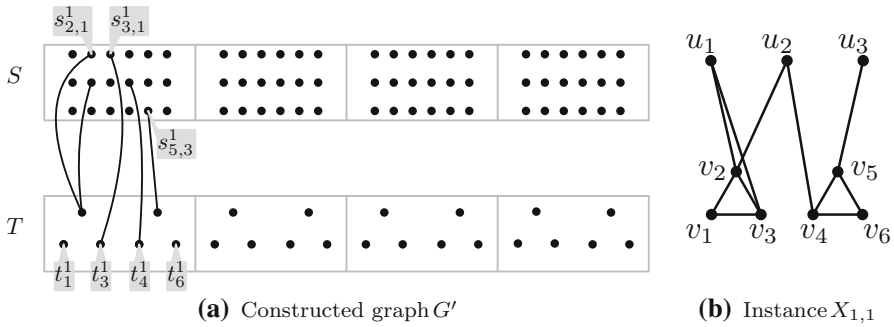


Fig. 1 Construction of graph G' for $t' = 4$, $m = 3$, and $n = 2$. Edges between vertices in S and T are shown for instance $X_{1,1}$. All blocking-gadgets and the vertex sets A and B are left out

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 Fig. 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 $s_{k,\ell}^i$ for $i \in [t']$, $k \in [3n]$ and $\ell \in [m]$. Define $L(s_{k,\ell}^i) := \{1, 2\}$. The vertices $s_{1,\ell}^i, s_{2,\ell}^i, \dots, s_{3n,\ell}^i$ 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 Fig. 1a).
2. Add t' sets of $3n$ vertices each, labeled T_j for $j \in [t']$. Label the vertices in T_j as t_k^j for $k \in [3n]$ and let $L(t_k^j) := \{1, 2, 3\}$. Vertices $t_{3k-2}^j, t_{3k-1}^j, t_{3k}^j$ 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 $s_{k,\ell}^i$ to vertex t_k^j 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 turning each triple $t_{3k-2}^j, t_{3k-1}^j, t_{3k}^j$ into a triangle for $k \in [n]$, and for each $\ell \in [m]$ merging all $3n$ vertices $s_{1,\ell}^i, \dots, s_{3n,\ell}^i$ into a single vertex u_ℓ .
4. Add vertex sets $A = \{a_1, \dots, a_{t'}\}$ and $B := \{b_1, \dots, b_{t'}\}$. 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 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 $s_{k,\ell}^i$ to π_1 , $s_{k+1,\ell}^i$ to π_2 , and a_i to π_3 . This ensures that when a_i has color 1, vertices $s_{k,\ell}^i$ and $s_{k',\ell}^i$ 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_{3k-2}^j to π_1 , t_{3k-1}^j to π_2 , t_{3k}^j 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 colored with three distinct colors. 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 3 *Let c be a 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 $c(s_{k,\ell}^i) = c(s_{k',\ell}^i)$.*

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 $c(s_{k,\ell}^i) \neq c(s_{k',\ell}^i)$. Then there must also exist $k \in [3n - 1]$ such that $c(s_{k,\ell}^i) \neq c(s_{k+1,\ell}^i)$, or else they would all be equal. Let (c_1, c_2, c_3) correspond to the coloring of $s_{k,\ell}^i, s_{k+1,\ell}^i$, and a_i as given by c , recall that by this definition $c_3 = 1$. Then blocking-gadget $((c_1, c_2, c_3))$ was added in Step 7 and connected to these three vertices. But by Lemma 11, 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 $s_{k,\ell}^i, s_{k+1,\ell}^i$ and a_i , one edge has two endpoints of equal color, which is a contradiction. \square

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 4 *Let c be a 3-list coloring of G' . Then there exists $j \in [t']$ such that for all $k \in [n]$ the triple $t_{3k}^j, t_{3k-1}^j, t_{3k-2}^j$ 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_{3k}^j, t_{3k-1}^j , and t_{3k-2}^j are not a colorful triple. Let $(c_1, c_2, c_3, c_4) \in [3]^4$ correspond to the coloring given to $t_{3k}^j, t_{3k-1}^j, t_{3k-2}^j$, 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 11, 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 $t_{3k}^j, t_{3k-1}^j, t_{3k-2}^j$, and b_j , one edge has two endpoints of equal color, which is a contradiction. \square

Claim 5 *The graph G' is 3-list colorable if and only if some input instance $X_{i^*j^*}$ is 2-3-colorable.*

Proof (\Rightarrow) Suppose we are given a 3-list coloring c of G' . By Claims 3 and 4 there exist integers i^* and $j^* \in [t']$ such that for all $\ell \in [m]$ and for all $k, k' \in [3n]$ we have $c(s_{k,\ell}^{i^*}) = c(s_{k',\ell}^{i^*})$ and furthermore for all $k \in [n]$ the triple $t_{3k}^{j^*}, t_{3k-1}^{j^*}, t_{3k-2}^{j^*}$ is colorful. We show that this implies that G_{i^*,j^*} has a valid 2-3-coloring c' , which we define as follows. Let $c'(u_\ell) := c(s_{1,\ell}^{i^*})$ for $\ell \in [m]$ and let $c'(v_k) := c(t_k^{j^*})$ 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

$$c'(v_k) = c(t_k^{j^*}) \neq c(s_{k,\ell}^{i^*}) = c(s_{1,\ell}^{i^*}) = c'(u_\ell).$$

For an edge $\{v_k, v'_k\} \in G_{i^*,j^*}$, its coloring corresponds to the coloring of $t_k^{j^*}$ and $t_{k'}^{j^*}$, which are colored differently by choice of j^* and Claim 4. Furthermore, u_ℓ is always colored with color 1 or 2 as $L(s_{1,\ell}^{i^*}) = \{1, 2\}$. Thereby, c' is indeed a 2-3-coloring of G_{i^*,j^*} .

(\Leftarrow) Suppose c is a 2-3-coloring of G_{i^*,j^*} , such that the partite set U 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 each $\ell \in [m]$, for each $k \in [3n]$, let $c'(s_{k,\ell}^{i^*}) := c(u_\ell)$. For $k \in [3n]$ let $c'(t_k^{j^*}) := c(v_k)$. For $j \neq j^*$ and $k \in [3n]$ let $c'(t_k^j) := 3$. For $i \neq i^* \in [t'], k \in [3n]$ and $\ell \in [m]$, pick $c'(s_{k,\ell}^i) \in \{1, 2\} \setminus \{c'(t_k^{j^*})\}$. 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 a 3-coloring on $G'[S \cup T]$. This will imply that the coloring defined so far is correct, 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 correct 3-coloring, it follows that $c'(s) \neq c'(t)$.

To complete the proof, extend c' to also 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 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^*} . In this second case the vertices $s_{k,\ell}^{i^*}$ and $s_{k+1,\ell}^{i^*}$ 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 11.

Similarly, coloring c' can be extended to blocking-gadgets(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. \square

The claim above shows that the constructed 3-LIST COLORING instance G' acts as a logical OR of the given input instances. 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 3-coloring if and only if the original graph had a 3-list coloring. Thus, by Claim 5, 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 two 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 3 now follows from Theorem 1 and Lemma 12. \square

The set of all vertices of a graph is always a valid vertex cover for that graph. Thereby, it follows from Theorem 3 that the lower bound also holds when parameterized by vertex cover. In [13, 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 $\text{NP} \subseteq \text{coNP/poly}$. Combining these results gives a lower bound for q -COLORING parameterized by vertex cover size.

Corollary 2 *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 $\text{NP} \subseteq \text{coNP/poly}$.*

The above lower bound carries over to q -COLORING parameterized by the size of a twin-cover, since any vertex cover of a graph is also a valid twin-cover. Recall that q -COLORING is equivalent to K_q -COLORING and $\Delta(K_q) = q - 1$. Thereby, the above lower bound matches the kernel size given in Theorem 2 up to $k^{o(1)}$ factors.

5 Conclusion

We have given a kernel for H -COLORING PARAMETERIZED BY TWIN-COVER with $\mathcal{O}(k^{\Delta(H)})$ vertices and bitsize $\mathcal{O}(k^{\Delta(H)} \log k)$. This kernel can be obtained without using information about (an approximation of) the minimum twin-cover of the input graph. It follows from this result that q -COLORING PARAMETERIZED BY VERTEX COVER has a kernel of 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 $\text{NP} \subseteq \text{coNP/poly}$. It was already known that for $q \geq 4$, q -COLORING PARAMETERIZED BY VERTEX COVER was unlikely to have 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 $\text{NP} \not\subseteq \text{coNP/poly}$. 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 twin-cover of the graph. This only changes the size of the obtained kernel by a constant factor. The kernel does not extend to the general LIST H -COLORING problem, since the gadget to simulate the list constraints only works correctly when H is a clique.

In this paper we gave a first example where finding redundant vertices and edges is done using appropriate polynomial equalities. 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.

Acknowledgements We thank Tim Hartmann for suggesting the use of twin-cover as a parameter. We are grateful to an anonymous referee of *Algorithmica* for suggesting the interpretation of the polynomial of Lemma 5 as a determinant.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Bodlaender, H.L., Jansen, B.M.P., Kratsch, S.: Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.* **28**(1), 277–305 (2014). <https://doi.org/10.1137/120880240>
2. Bougeret, M., Sau, I.: How much does a treedepth modulator help to obtain polynomial kernels beyond sparse graphs? In: Proceedings of 12th IPEC, pp. 10:1–10:13 (2017). <https://doi.org/10.4230/LIPIcs.IPEC.2017.10>
3. Chitnis, R., Egri, L., Marx, D.: List H -coloring a graph by removing few vertices. *Algorithmica* **78**(1), 110–146 (2017). <https://doi.org/10.1007/s00453-016-0139-6>
4. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: *Parameterized Algorithms*. Springer, Berlin (2015). <https://doi.org/10.1007/978-3-319-21275-3>
5. Dell, H., Marx, D.: Kernelization of packing problems. In: Proceedings of 23th SODA, SODA '12, pp. 68–81 (2012)
6. Dell, H., van Melkebeek, D.: Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM* **61**(4), 23:1–23:27 (2014). <https://doi.org/10.1145/2629620>
7. Downey, R.G., Fellows, M.R.: *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, Berlin (2013). <https://doi.org/10.1007/978-1-4471-5559-1>
8. Fiala, J., Golovach, P.A., Kratochvíl, J.: Parameterized complexity of coloring problems: treewidth versus vertex cover. *Theor. Comput. Sci.* **412**(23), 2513–2523 (2011). <https://doi.org/10.1016/j.tcs.2010.10.043>
9. Ganian, R.: Improving vertex cover as a graph parameter. *Discrete Math. Theor. Comput. Sci.* **17**(2), 77–100 (2015)
10. Hell, P., Nešetřil, J.: On the complexity of H -coloring. *J. Comb. Theory Ser. B* **48**(1), 92–110 (1990). [https://doi.org/10.1016/0095-8956\(90\)90132-J](https://doi.org/10.1016/0095-8956(90)90132-J)
11. Hols, E.M.C., Kratsch, S.: Smaller parameters for vertex cover kernelization. In: Proceedings of 12th IPEC, pp. 20:1–20:12 (2017). <https://doi.org/10.4230/LIPIcs.IPEC.2017.20>
12. Jaffke, L., Jansen, B.M.P.: Fine-grained parameterized complexity analysis of graph coloring problems. In: Proceedings of 10th CIAC, pp. 345–356 (2017). https://doi.org/10.1007/978-3-319-57586-5_29
13. Jansen, B.M.P., Kratsch, S.: Data reduction for graph coloring problems. *Inf. Comput.* **231**, 70–88 (2013). <https://doi.org/10.1016/j.ic.2013.08.005>

14. Jansen, B.M.P., Pieterse, A.: Optimal sparsification for some binary CSPs using low-degree polynomials. In: Proceedings of 41st MFCS, pp. 71:1–71:14 (2016). <https://doi.org/10.4230/LIPIcs.MFCS.2016.71>
15. Jansen, B.M.P., Pieterse, A.: Sparsification upper and lower bounds for graph problems and not-all-equal SAT. *Algorithmica* **79**(1), 3–28 (2017). <https://doi.org/10.1007/s00453-016-0189-9>
16. Kratsch, S., Philip, G., Ray, S.: Point line cover: the easy kernel is essentially tight. *ACM Trans. Algorithms* **12**(3), 40:1–40:16 (2016). <https://doi.org/10.1145/2832912>
17. Spinrad, J.P.: *Efficient Graph Representations*. American Mathematical Society, Providence (2003)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.