

PARAMETERIZED COMPLEXITY OF CONFLICT-FREE GRAPH COLORING*

HANS L. BODLAENDER[†], SUDESHNA KOLAY[‡], AND ASTRID PIETERSE[§]

Abstract. Given a graph G , a q -open neighborhood conflict-free coloring or q -ONCF-coloring is a vertex coloring $c: V(G) \rightarrow \{1, 2, \dots, q\}$ such that for each vertex $v \in V(G)$ there is a vertex in $N(v)$ that is uniquely colored from the rest of the vertices in $N(v)$. When we replace $N(v)$ by the closed neighborhood $N[v]$, then we call such a coloring a q -closed neighborhood conflict-free coloring or simply q -CNCF-coloring. In this paper, we study the NP-hard decision questions of whether for a constant q an input graph has a q -ONCF-coloring or a q -CNCF-coloring. We will study these two problems in the parameterized setting. First of all, we study running time bounds on fixed-parameter tractable algorithms for these problems when parameterized by treewidth. We improve the existing upper bounds, and also provide lower bounds on the running time under the exponential time hypothesis and the strong exponential time hypothesis. Second, we study the kernelization complexity of both problems, using vertex cover as the parameter. We show that both $(q \geq 2)$ -ONCF-coloring and $(q \geq 3)$ -CNCF-coloring cannot have polynomial kernels when parameterized by the size of a vertex cover unless $\text{NP} \subseteq \text{coNP/poly}$. On the other hand, we obtain a polynomial kernel for 2-CNCF-coloring parameterized by vertex cover. We conclude the study with some combinatorial results. Denote $\chi_{ON}(G)$ and $\chi_{CN}(G)$ to be the minimum number of colors required to ONCF-color and CNCF-color G , respectively. Upper bounds on $\chi_{CN}(G)$ with respect to structural parameters like minimum vertex cover size, minimum feedback vertex set size, and treewidth are known. To the best of our knowledge only an upper bound on $\chi_{ON}(G)$ with respect to minimum vertex cover size was known. We provide tight bounds for $\chi_{ON}(G)$ with respect to minimum vertex cover size. Also, we provide the first upper bounds on $\chi_{ON}(G)$ with respect to minimum feedback vertex set size and treewidth.

Key words. conflict-free coloring, kernelization, fixed-parameter tractability, combinatorial bounds

AMS subject classifications. 68Q25, 05C15

DOI. 10.1137/19M1307160

1. Introduction. Often, in frequency allocation problems for cellular networks, it is important to allot a unique frequency for each client, so that at least one frequency is unaffected by cancellation. Such problems can be theoretically formulated as a coloring problem on a set system, better known as conflict-free coloring [8]. Formally, given a set system $\mathcal{H} = (U, \mathcal{F})$, a q -conflict-free coloring $c: U \rightarrow \{1, 2, \dots, q\}$ is a function where for each set $f \in \mathcal{F}$, there is an element $v \in f$ such that for all $w \neq v \in f$, $c(v) \neq c(w)$. In other words, each set f has at least one element that is uniquely colored in the set. This variant of coloring has also been extensively studied for set systems induced by various geometric regions [2, 13, 21].

*Received by the editors December 16, 2019; accepted for publication (in revised form) May 5, 2021; published electronically September 2, 2021. An extended abstract of this work appeared under the same title in the proceedings of the 16th Workshop on Algorithms and Data Structures, WADS 2019, Springer, Cham, Switzerland, 2019, pp. 168–180. The research was partially done when all three authors were associated with Eindhoven University of Technology. Part of this work was done at the Lorentz center workshop on Fixed-Parameter Computational Geometry, May 14–18, 2018, in Leiden, the Netherlands.

<https://doi.org/10.1137/19M1307160>

Funding: This research was done with support by the NWO Gravitation grant NETWORKS.

[†]Department of Computer Science, Utrecht University, Utrecht, 3508 TB, The Netherlands (h.l.bodlaender@uu.nl).

[‡]Indian Institute of Technology, Kharagpur, India (skolay@cse.iitkgp.ac.in).

[§]Department of Computer Science, Humboldt Universität zu Berlin, Berlin, 10117 Germany (astridpieterse@outlook.com).

A natural step to study most coloring problems is to study them in graphs. Given a graph G , $V(G)$ denotes the set of n vertices of G while $E(G)$ denotes the set of m edges in G . A q -coloring of G , for $q \in \mathbb{N}$ is a function $c: V(G) \rightarrow \{1, 2, \dots, q\}$. The most well-studied coloring problem on graphs is proper-coloring. A q -coloring c is called a proper-coloring if for each edge $\{u, v\} \in E(G)$, $c(u) \neq c(v)$. In this paper, we study two specialized variants of q -conflict-free coloring on graphs, known as q -ONCF-coloring and q -CNCF-coloring, which are defined as follows.

DEFINITION 1.1. *Given a graph G , a q -coloring $c: V(G) \rightarrow \{1, 2, \dots, q\}$ is called a q -ONCF-coloring, if for every vertex $v \in V(G)$, there is a vertex u in the open neighborhood $N(v)$ such that $c(u) \neq c(w)$ for all $w \neq u \in N(v)$. In other words, every open neighborhood in G has a uniquely colored vertex.*

DEFINITION 1.2. *Given a graph G , a q -coloring $c: V(G) \rightarrow \{1, 2, \dots, q\}$ is called a q -CNCF-coloring, if for every vertex $v \in V(G)$, there is a vertex u in the closed neighborhood $N[v]$ such that $c(u) \neq c(w)$ for all $w \neq u \in N[v]$. In other words, every closed neighborhood in G has a uniquely colored vertex.*

Observe that by the above definitions, the q -ONCF-coloring (or q -CNCF-coloring) problem is a special case of the conflict-free coloring of set systems. Given a graph G , we can associate it with the set system $\mathcal{H} = (V(G), \mathcal{F})$, where \mathcal{F} consists of the sets given by open neighborhoods $N(v)$ (respectively, closed neighborhoods $N[v]$) for $v \in V(G)$. A q -ONCF-coloring (or q -CNCF-coloring) of G then corresponds to a q -conflict-free coloring of the associated set system.

Notationally, let $\chi_{CF}(\mathcal{H})$ denote the minimum number of colors required for a conflict-free coloring of a set system \mathcal{H} . Similarly, we denote by $\chi_{ON}(G)$ and $\chi_{CN}(G)$ the minimum number of colors required for an ONCF-coloring and a CNCF-coloring of a graph G , respectively. The study of conflict-free coloring was initially restricted to combinatorial studies. This was first explored in [8] and [20]. Pach and Tardos [18] gave an upper bound of $\mathcal{O}(\sqrt{m})$ on $\chi_{CF}(\mathcal{H})$ for a set system $\mathcal{H} = (U, \mathcal{F})$ when the size of \mathcal{F} is m . In [18], it was also shown that for a graph G with n vertices $\chi_{CN}(G) = \mathcal{O}(\log^2 n)$. This bound was shown to be tight in [12]. Similarly, [5] showed that $\chi_{ON}(G) = \Theta(\sqrt{n})$.

However, computing $\chi_{ON}(G)$ or $\chi_{CN}(G)$ is NP-hard. This is because deciding whether a 2-ONCF-coloring or a 2-CNCF-coloring of G exists is NP-hard [11]. This motivates the study of the following decision problems under the lens of parameterized complexity.

q -ONCF-COLORING

Input: A graph G .

Question: Is there a q -ONCF-coloring of G ?

The q -CNCF-COLORING problem is defined analogously.

Note that because of the NP-hardness for q -ONCF-COLORING or q -CNCF-COLORING even when $q = 2$, the two problems are para-NP-hard under the natural parameter q . Thus, the problems were studied under structural parameters. Gargano and Rescigno [11] showed that both q -ONCF-COLORING and q -CNCF-COLORING have fixed-parameter tractable (FPT) algorithms when parameterized by (i) the size of a vertex cover of the input graph G , and (ii) the neighborhood diversity of the input graph. Gargano and Rescigno also mention that due to Courcelle's theorem, for a nonnegative constant q , the two decision problems are FPT with the treewidth of the input graph as the parameter.

Our Results and Contributions. In this paper, we extend the parameterized study of the above two problems with respect to structural parameters. Our first objective is to provide both upper and lower bounds for FPT algorithms when using treewidth as the parameter (section 3). We show that both q -ONCF-COLORING and q -CNCF-COLORING parameterized by treewidth t can be solved in time $(2q^2)^t n^{\mathcal{O}(1)}$. On the other hand, for $q \geq 3$, both problems cannot be solved in time $(q - \varepsilon)^t n^{\mathcal{O}(1)}$ under the strong exponential time hypothesis (SETH). For $q = 2$, both problems cannot be solved in time $2^{o(t)} n^{\mathcal{O}(1)}$ under exponential time hypothesis (ETH).

We also study the polynomial kernelization question (section 4). Observe that both q -ONCF-COLORING and q -CNCF-COLORING cannot have polynomial kernels under treewidth as the parameter (assuming $\text{NP} \not\subseteq \text{coNP/poly}$), as there are straightforward AND-cross-compositions from each problem to itself.¹ Therefore, we will study the kernelization question by a larger parameter, namely, the size of a vertex cover in the input graph. The kernelization complexity of the q -COLORING problem (asking for a proper-coloring of the input graph) is very well studied for this parameter; the problem admits a kernel of size $\tilde{\mathcal{O}}(k^{q-1})$ [15] which is known to be tight unless $\text{NP} \subseteq \text{coNP/poly}$ [14]. From this perspective however, q -CNCF-COLORING and q -ONCF-COLORING turn out to be much harder: q -CNCF-COLORING for $q \geq 3$ and q -ONCF-COLORING for $q \geq 2$ do not have polynomial kernels under the standard complexity assumptions, when parameterized by the size of a vertex cover. Interestingly, 2-CNCF-COLORING parameterized by vertex cover size *does* have a polynomial kernel and we obtain an explicit polynomial compression for the problem. Although this does not lead to a polynomial kernel of reasonable size, we study a restricted version called 2-CNCF-COLORING-VC-EXTENSION (Section 4.4) and show that this problem has an $\mathcal{O}(k^2 \log k)$ kernel where k is the vertex cover size. Therefore, 2-CNCF-COLORING behaves significantly differently from the other problems.

Finally, we obtain a number of combinatorial results regarding ONCF-colorings of graphs. Denote by $\chi(G)$ the minimum q for which a q -proper-coloring for G exists. While $\chi_{CN}(G) \leq \chi(G)$, the same upper bound does not hold for $\chi_{ON}(G)$ [11]. For a graph G , let $\text{vc}(G)$, $\text{fvs}(G)$, and $\text{tw}(G)$ denote the size of a minimum vertex cover, the size of a minimum feedback vertex set, and the treewidth of G , respectively. From the known result that $\chi(G) \leq \text{tw}(G) + 1 \leq \text{fvs}(G) + 1 \leq \text{vc}(G) + 1$, we could immediately obtain the fact that the same behavior holds for $\chi_{CN}(G)$. However, to show that $\chi_{ON}(G)$ behaves similarly more work needs to be done. To the best of our knowledge no upper bounds on $\chi_{ON}(G)$ with respect to $\text{fvs}(G)$ and $\text{tw}(G)$ were known, while a loose upper bound was provided with respect to $\text{vc}(G)$ in [11]. We give a tight upper bound on $\chi_{ON}(G)$ with respect to $\text{vc}(G)$ and also provide the first upper bounds on $\chi_{ON}(G)$ with respect to $\text{fvs}(G)$ and $\text{tw}(G)$ (section 5).

Our main contributions in this work are structural results for the conflict-free coloring problem, which we believe gives more insight into the decision problems on graphs. First, the gadgets we build for the ETH-based lower bounds could be useful for future lower bounds, but are also useful for understanding difficult examples for conflict-free coloring which have not been known in abundance so far. We are able to reuse these gadgets in the constructions needed to prove the kernelization lower bounds. Second, our combinatorial results also give constructible conflict-free colorings of graphs and therefore provide more insight into conflict-free colored graphs.

¹This is true for a number of graph problems when parameterized by treewidth. For more information, see [6, Theorem 15.12] and the example given for TREEWIDTH (parameterized by solution size) in [6, page 534].

Finally, the kernelization dichotomy we obtain for q -ONCF-COLORING and q -CNCF-COLORING under vertex cover size as a parameter is a very surprising one.

2. Preliminaries. For a positive integer n , we denote the set $\{1, 2, \dots, n\}$ in short by $[n]$. For a graph G , given a q -coloring $c: V(G) \rightarrow [q]$ and a subset $S \subseteq V(G)$, we denote by $c|_S$ the restriction of c to the subset S . For a graph G that is q -ONCF-colored by a coloring c , for a vertex $v \in V(G)$, suppose $w \in N(v)$ is such that $c(w) \neq c(w')$ for each $w' \neq w \in N(v)$; then $c(w)$ is referred to as the ONCF-color of v . Similarly, for a graph G that is q -CNCF-colored by a coloring c , for a vertex $v \in V(G)$, a unique color in $N[v]$ is referred to as the CNCF-color of v .

An *edge-star graph* is a generalization of a star graph where there is a central edge $\{u, v\}$ and all other vertices w have $N(w) = \{u, v\}$. A triangle is an example of an edge-star graph.

In some of our lower bound constructions, we will use the following useful gadgets, to construct our color palette. These gadget were first defined by Abel et al. [1, section 3.1].

DEFINITION 2.1 (see [1]). *For every positive integer k , a graph G_k is recursively defined as follows:*

1. G_1 consists of a single isolated vertex. G_2 is a $K_{1,3}$ with one edge subdivided by another vertex (refer also to Figure 4).
2. Given G_k and G_{k-1} , G_{k+1} is constructed as follows for $k \geq 2$:
 - Take a complete graph $G = K_{k+1}$ on $k+1$ vertices.
 - To each vertex $v \in V(K_{k+1})$, attach two disjoint and independent copies of G_k , adding an edge from v to every vertex of both copies of G_k .
 - For each edge $e = \{v, w\} \in E(K_{k+1})$, add two disjoint and independent copies of G_{k-1} , adding an edge from v and w to every vertex of both copies.

Let $\chi_{\text{CN}}^*(G)$ denote the minimum number of colors needed to CNCF-color G when it is allowed to not color certain vertices. Abel et al. have shown the following lemma.

LEMMA 2.2 (see [1, Lemma 3.3]). *For G_k constructed according to Definition 2.1, it holds that $\chi_{\text{CN}}^*(G_k) = k$.*

We also have the following fact about ONCF-coloring of trees.

OBSERVATION 2.3. *Let T be a tree. There is a polynomial time procedure that outputs a 2-ONCF-coloring of T .*

Proof. Let the color set be $\{0, 1\}$. Pick an arbitrary vertex r and root T at r . Let \mathcal{I} be the ordering on the vertices in $V(T)$ obtained from doing a breadth first search traversal of T starting at the root r . The following 2-coloring function $f: V(T) \rightarrow \{0, 1\}$ is defined on $V(T)$:

1. The root r is colored such that $f(r) = 0$.
2. Let v be the first child of r according to \mathcal{I} . Then we set $f(v) = 0$ and $\forall u \in N(r) \setminus \{v\}, f(u) = 1$.
3. Consider the smallest uncolored vertex $u \in V(T) \setminus N[r]$. We set $f(u) = 1 - f(u^g)$, where u^g is the (unique) grandparent of u .

From the description of the coloring it is clear that the procedure takes polynomial time. We claim that f 2-ONCF-colors T :

- The root r has a uniquely colored neighbor in its first child v according to \mathcal{I} .
- Due to point 3 of the definition of f , for any vertex $v \in V(T) \setminus \{r\}$, its parent

gets a color different from all its children. Therefore, the uniquely colored neighbor of v is its parent.

Thus, we are done. □

2.1. Tree decompositions and treewidth. We define treewidth and tree decompositions.

DEFINITION 2.4 (tree decomposition [6]). *A tree decomposition of an (undirected or directed) graph G is a tuple $\mathcal{T} = (T, \{X_{\mathbf{u}}\}_{\mathbf{u} \in V(T)})$, where T is a tree in which each vertex $\mathbf{u} \in V(T)$ has an assigned set of vertices $X_{\mathbf{u}} \subseteq V(G)$ (called a bag) such that the following properties hold:*

- $\bigcup_{\mathbf{u} \in V(T)} X_{\mathbf{u}} = V(G)$.
- For any $xy \in E(G)$, there exists a $\mathbf{u} \in V(T)$ such that $x, y \in X_{\mathbf{u}}$.
- If $x \in X_{\mathbf{u}}$ and $x \in X_{\mathbf{v}}$, then $x \in X_{\mathbf{w}}$ for all \mathbf{w} on the path from \mathbf{u} to \mathbf{v} in T .

In short, we denote $\mathcal{T} = (T, \{X_{\mathbf{u}}\}_{\mathbf{u} \in V(T)})$ as T .

The *treewidth* $tw_{\mathcal{T}}$ of a tree decomposition \mathcal{T} is the size of the largest bag of \mathcal{T} minus one. A graph may have several distinct tree decompositions. The treewidth $tw(G)$ of a graph G is defined as the minimum of treewidths over all possible tree decompositions of G . Note that for the tree T of a tree decomposition, we denote a vertex of $V(T)$ in bold font. If T is rooted at a vertex \mathbf{r} , for a vertex $\mathbf{u} \in V(T)$, $V_{\mathbf{u}} = \bigcup_{\mathbf{v} \in T'} X_{\mathbf{v}}$, where T' is the subtree rooted at \mathbf{u} .

A tree decomposition $\mathcal{T} = (T, \{X_{\mathbf{u}}\}_{\mathbf{u} \in V(T)})$ is called a *nice tree decomposition* if T is a tree rooted at some node \mathbf{r} where $X_{\mathbf{r}} = \emptyset$, each node of T has at most two children, and each node is of one of the following kinds:

Introduce node: a node \mathbf{u} that has only one child \mathbf{u}' , where $X_{\mathbf{u}} \supset X_{\mathbf{u}'}$ and $|X_{\mathbf{u}}| = |X_{\mathbf{u}'}| + 1$.

Forget vertex node: a node \mathbf{u} that has only one child \mathbf{u}' , where $X_{\mathbf{u}} \subset X_{\mathbf{u}'}$ and $|X_{\mathbf{u}}| = |X_{\mathbf{u}'}| - 1$.

Join node: a node \mathbf{u} with two children \mathbf{u}_1 and \mathbf{u}_2 such that $X_{\mathbf{u}} = X_{\mathbf{u}_1} = X_{\mathbf{u}_2}$.

Leaf node: a node \mathbf{u} that is a leaf of T , and $X_{\mathbf{u}} = \emptyset$.

One can show that a tree decomposition of width w can be transformed into a nice tree decomposition of the same width w and with $\mathcal{O}(w|V(G)|)$ nodes; see, e.g., [6].

We modify the definition of a nice tree decomposition slightly by ensuring that no bag in the tree decomposition is empty. This can easily be done by adding an arbitrary vertex $z \in V(G)$ to all bags of the current nice tree decomposition. This will ensure the nonemptiness property. Note that our nice tree decomposition will have width $w + 1$.

2.2. Parameterized complexity. Let Σ be a finite alphabet. A parameterized problem \mathcal{Q} is a subset of $\Sigma^* \times \mathbb{N}$.

DEFINITION 2.5 (kernelization). *Let $\mathcal{Q}, \mathcal{Q}'$ be two parameterized problems and let $h: \mathbb{N} \rightarrow \mathbb{N}$ be some computable function. A generalized kernel from \mathcal{Q} to \mathcal{Q}' of size $h(k)$ is an algorithm that given an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, outputs $(x', k') \in \Sigma^* \times \mathbb{N}$ in time $\text{poly}(|x| + k)$ such that (i) $(x, k) \in \mathcal{Q}$ if and only if $(x', k') \in \mathcal{Q}'$, and (ii) $|x'| \leq h(k)$ and $k' \leq h(k)$.*

The algorithm is a kernel if $\mathcal{Q} = \mathcal{Q}'$. It is a polynomial (generalized) kernel if $h(k)$ is a polynomial in k .

Next, we describe a few methods that can be used to rule out the existence of polynomial kernels. One such method is by a polynomial parameter transformation

[4] from a problem that is known to not admit a polynomial kernel. We repeat the necessary information here for completeness.

DEFINITION 2.6 (polynomial parameter transformation [4]). *Let \mathcal{Q} and \mathcal{Q}' be parameterized problems. A polynomial parameter transformation from \mathcal{Q} to \mathcal{Q}' is an algorithm that takes an input (x, k) and outputs (x', k') such that the following hold.*

- $(x, k) \in \mathcal{Q}$ if and only if $(x', k') \in \mathcal{Q}'$, and
- k' is bounded by a polynomial in k .

We denote this as $\mathcal{Q} \leq_{\text{ppt}} \mathcal{Q}'$.

The following theorem follows from [3, Prop. 2.16] and shows how to obtain lower bounds using polynomial parameter transformations.

THEOREM 2.7 (see [3]). *Let \mathcal{Q} and \mathcal{Q}' be parameterized problems with $\mathcal{Q} \leq_{\text{ppt}} \mathcal{Q}'$. If \mathcal{Q}' admits a polynomial generalized kernel, then \mathcal{Q} admits a polynomial generalized kernel.*

Another way to rule out the existence of polynomial kernels is using the framework of cross-compositions [3]. We start by providing the necessary definitions.

DEFINITION 2.8 (polynomial equivalence relation [3]). *An equivalence relation \mathcal{R} on Σ^* is called a polynomial equivalence relation if the following two 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 2.9 (cross-composition [3]). *Let $L \subseteq \Sigma^*$ be a language, let \mathcal{R} be a polynomial equivalence relation on Σ^* , and let $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. An OR-cross-composition of L into \mathcal{Q} (with respect to \mathcal{R}) 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 following hold:*

- The parameter value k is polynomially bounded in $\max_{i=1}^t |x_i| + \log t$.
- The instance (y, k) is a yes-instance for \mathcal{Q} if and only if at least one instance x_i is a yes-instance for L .

The following theorem shows how cross-compositions are used to prove kernelization lower bounds.

THEOREM 2.10 (see [3]). *If an NP-hard language L OR-cross-composes into the parameterized problem \mathcal{Q} , then \mathcal{Q} does not admit a (generalized) polynomial kernelization unless $\text{NP} \subseteq \text{coNP}/\text{poly}$.*

2.3. Fast subset convolution computation. Given a universe U with n elements, the subset convolution of two functions $f, g : 2^U \rightarrow \mathbb{Z}$ is a function $(f * g) : 2^U \rightarrow \mathbb{Z}$ such that for every $Y \subseteq U$, $(f * g)(Y) = \sum_{X \subseteq Y} f(X)g(Y - X)$. Equivalently, $(f * g)(Y) = \sum_{A \oplus B = Y} f(A)g(B)$.

PROPOSITION 2.11 (see [9]). *For two functions $f, g : 2^U \rightarrow \mathbb{Z}$, given all the 2^n values of f and g in the input, all the 2^n values of the subset convolution $f * g$ can be computed in $\mathcal{O}(2^n \cdot n^3)$ arithmetic operations.*

In fact, this result can be extended to subset convolution of functions that map to

any ring, instead of $(\mathbb{Z}, +, \times)$ [6]. Consider the set $\mathbb{Z} \cup \{-\infty\}$, with the added relation that $\forall z \in \mathbb{Z}, \{-\infty\} < z$. The *max* operator takes two elements from this set and outputs the maximum of the two elements. Notice that $\mathbb{Z} \cup \{-\infty\}$, along with *max* as an additive operator and $+$ as a multiplicative operator, forms a semiring [6]. We will call this semiring the integer max-sum semiring. The subset convolution of two functions $f, g : 2^U \rightarrow \mathbb{Z} \cup \{-\infty\}$ with *max* and $+$ as the additive and multiplicative operators, becomes $(f * g)(Y) = \max_{A \uplus B = Y} f(A) + g(B)$.

PROPOSITION 2.12 (see [9]). *Given two functions $f, g : 2^U \rightarrow \{-M, \dots, M\}$, all the 2^n values of f and g in the input, and all the 2^n values of the subset convolution $(f * g)$ over the integer max-sum semiring can be computed in time $2^n n^{\mathcal{O}(1)} \cdot \mathcal{O}(M \log M \log \log M)$.*

For more details about subset convolutions and fast calculations of subset convolutions, please refer to [6, 9].

3. Algorithmic results parameterized by treewidth. In this section, we state the algorithmic results obtained for the ONCF-COLORING and CNCF-COLORING problems parameterized by treewidth. On the algorithmic side, we have the following theorem.

THEOREM 3.1. *q -ONCF-COLORING and q -CNCF-COLORING parameterized by treewidth t admit a $(2q^2)^t n^{\mathcal{O}(1)}$ time algorithm.*

We also obtain algorithmic lower bounds for the problems under standard assumptions.

THEOREM 3.2. *The following algorithmic lower bounds can be obtained:*

1. *For $q \geq 3$, q -ONCF-COLORING or q -CNCF-COLORING parameterized by treewidth t cannot be solved in $(q - \epsilon)^t n^{\mathcal{O}(1)}$ time under SETH.*
2. *2-ONCF-COLORING or 2-CNCF-COLORING parameterized by treewidth t cannot be solved in $2^{\mathcal{O}(t)} n^{\mathcal{O}(1)}$ time under ETH.*

In the remainder of this section, we will prove the two theorems stated above.

3.1. Algorithms. In this section, we prove Theorem 3.1. In the following lemma, we describe an algorithm for q -ONCF-COLORING, parameterized by treewidth. The algorithm for q -CNCF-COLORING parameterized by treewidth is very similar and has the same running time.

LEMMA 3.3. *q -ONCF-COLORING parameterized by treewidth t admits a $(2q^2)^t n^{\mathcal{O}(1)}$ -time algorithm.*

Proof. We assume that a nice tree decomposition $\mathcal{T} = (T, \{X_{\mathbf{u}}\}_{\mathbf{u} \in V(T)})$, rooted at a leaf \mathbf{r} , is given to us. Also, recall that no bag is empty, and that each leaf bag or the root bag has exactly one vertex in it. We proceed with the following treewidth dynamic programming. Given a bag $X_{\mathbf{i}}$ corresponding to the vertex $\mathbf{i} \in V(T)$, a state for the bag is a tuple $(\mathbf{i}, c_1, c_2, f)$, where

- \mathbf{i} determines the bag;
- $c_1 : X_{\mathbf{i}} \rightarrow [q]$ is a vertex coloring of $X_{\mathbf{i}}$. Intuitively, for a vertex $x \in X_{\mathbf{i}}$, $c_1(x)$ is the color x receives in the conflict-free coloring we are after;
- $c_2 : X_{\mathbf{i}} \rightarrow [q]$ is a color assignment to each vertex of $X_{\mathbf{i}}$. For a vertex $x \in X_{\mathbf{i}}$, $c_2(x)$ should be the color that occurs exactly once in the neighborhood of x ;
- $f : X_{\mathbf{i}} \rightarrow \{0, 1\}$ is an indicator function for the vertices of $X_{\mathbf{i}}$. The idea is that $f(x)$ indicates whether x already has a neighbor of color $c_2(x)$ in the subtree rooted at \mathbf{i} .

Let \mathcal{S}_i be the set of all states associated with X_i . A function $\Gamma_i: \mathcal{S}_i \rightarrow \{0, 1\}$ is defined as follows: For a state $s = (\mathbf{i}, c_1, c_2, f)$, suppose there is a vertex coloring $c: V_i \rightarrow [q]$ such that (i) its restriction to the vertices in X_i is the coloring c_1 ; (ii) for each $v \in X_i$, the color $c_2(v)$ is used at most once in $N(v) \cap V_i$; (iii) for each $v \in X_i$, if there is a vertex $w \in N(v) \cap V_i$ such that $c_1(w) = c_2(v)$, then $f(v) = 1$ and otherwise $f(v) = 0$; (iv) for any vertex $v \in V_i \setminus X_i$, $N(v) \subseteq V_i$ has a uniquely colored vertex under coloring c . Then $\Gamma_i(s) = 1$. Otherwise, $\Gamma_i(s) = 0$. In other words, c is such that except for the vertices in X_i the graph induced on V_i is ONCF-colored and a state s stores a snapshot of c at the boundary X_i of the graph seen so far.

Our dynamic programming will calculate the function Γ_i for each bag \mathbf{i} . Note that for the root $X_r = \{z\}$, if in \mathcal{S}_r there is a state $(\mathbf{r}, c_1, c_2, f)$ such that $f(z) = 1$ and $\Gamma_r(s) = 1$, then the graph G has a q -ONCF-coloring. We describe our dynamic programming in cases according to the types of nodes of the tree decomposition.

Leaf node: Let $X_i = \{z\}$ be a leaf node. Then,

$$\Gamma_i(\mathbf{i}, c_1, c_2, f) = \begin{cases} 1 & \text{if } f(z) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

This can be calculated in $2q^2$ time. For the correctness, note that the uniquely colored neighbor of z cannot appear in the graph seen so far as a leaf node only contains z .

Forget node: Let X_i be a forget node with its child being X_j . Also, let $X_j = X_i \cup \{v\}$. Consider a state $s = (\mathbf{i}, c_1, c_2, f) \in \mathcal{S}_i$ and a state $s' = (\mathbf{j}, d_1, d_2, g) \in \mathcal{S}_j$. We say that s' is consistent with s , or $s' \leq_c s$ if (i) $d_1|_{X_i} = c_1, d_2|_{X_i} = c_2$, (ii) $g|_{X_i} = f$ and $g(v) = 1$. Then,

$$\Gamma_i(\mathbf{i}, c_1, c_2, f) = \begin{cases} \Gamma_j(\mathbf{j}, d_1, d_2, g) & \text{if } (\mathbf{j}, d_1, d_2, g) \leq_c (\mathbf{i}, c_1, c_2, f), \\ 0 & \text{otherwise.} \end{cases}$$

This can be calculated in $(2q^2)^{|X_i|}$ time. To prove correctness, first suppose that $\Gamma_i(\mathbf{i}, c_1, c_2, f) = 1$ and let $c: V_i \rightarrow [q]$ be a coloring that is a witness to this. By definition of consistency, there is only one state $(\mathbf{j}, d_1, d_2, g)$ such that $(\mathbf{j}, d_1, d_2, g) \leq_c (\mathbf{i}, c_1, c_2, f)$. Since $V_i = V_j$, the same coloring c also witnesses that $\Gamma_j(\mathbf{j}, d_1, d_2, g) = 1$. Conversely, suppose $\Gamma_j(\mathbf{j}, d_1, d_2, g) = 1$, where $(\mathbf{j}, d_1, d_2, g)$ is the unique state such that $(\mathbf{j}, d_1, d_2, g) \leq_c (\mathbf{i}, c_1, c_2, f)$. Let $c: V_j \rightarrow [q]$ be a coloring that is a witness to this. Since $V_i = V_j$ and by definition of consistency $g(v) = 1$, the same coloring c also witnesses the fact that $\Gamma_i(\mathbf{i}, c_1, c_2, f) = 1$. Thus, our recurrence correctly calculates $\Gamma_i(\mathbf{i}, c_1, c_2, f)$.

Introduce node: Let X_i be an introduce node with its child being X_j . Also, let $X_i = X_j \cup \{v\}$. Consider a state $s = (\mathbf{i}, c_1, c_2, f) \in \mathcal{S}_i$ and a state $s' = (\mathbf{j}, d_1, d_2, g) \in \mathcal{S}_j$. We say that s' is consistent with s , or $s' \leq_c s$ (i) if $c_1|_{X_j} = d_1, c_2|_{X_j} = d_2$; (ii) if there is a $w \in N(v) \cap X_i$ such that $c_2(v) = c_1(w)$ then there is exactly one such w and $f(v) = 1$, otherwise there is no such w and $f(v) = 0$; (iii) if there is a $w \in N(v) \cap X_i$ such that $c_1(v) = c_2(w)$ then $g(w) = 0$ and $f(w) = 1$; (iv) for all other $u \in X_j$, $f(u) = g(u)$. Then,

$$\Gamma_i(\mathbf{i}, c_1, c_2, f) = \begin{cases} \max_{s' \in \mathcal{S}_j} \Gamma_j(s') & \text{such that } s' \leq_c (\mathbf{i}, c_1, c_2, f), \\ 0 & \text{otherwise.} \end{cases}$$

This can be calculated in $(2q^2)^{|X_i|}$ time. To prove correctness, first suppose that $\Gamma_i(\mathbf{i}, c_1, c_2, f) = 1$ and let $c: V_i \rightarrow [q]$ be a coloring that is a witness to this. By the

definition of consistency, there is at least one state s' such that $s' \leq_c (\mathbf{i}, c_1, c_2, f)$ and the same coloring c also witnesses the fact that $\Gamma_j(s') = 1$. Conversely, suppose there is a state s' such that $s' \leq_c (\mathbf{i}, c_1, c_2, f)$ and $\Gamma_j(s') = 1$. Then by definition of consistency, $\Gamma_i(\mathbf{i}, c_1, c_2, f) = 1$. Thus, our recurrence correctly calculates $\Gamma_i(\mathbf{i}, c_1, c_2, f)$.

Join node: Let X_i be a join node with its children being X_a and X_b . This means that $X_i = X_a = X_b$. Consider a state $s = (\mathbf{i}, c_1, c_2, f) \in \mathcal{S}_i$, and states $s' = (\mathbf{a}, d_1, d_2, g) \in \mathcal{S}_a, s'' = (\mathbf{b}, e_1, e_2, h) \in \mathcal{S}_b$. We say that $\{s', s''\}$ is consistent with s , or $\{s', s''\} \leq_c s$; (i) if $c_1 = d_1 = e_1, c_2 = d_2 = e_2$; (ii) if there is a $v \in X_i$ such that $g(v) = 1$ ($h(v) = 1$) then $h(v) = 0$ ($g(v) = 0$) and $f(v) = 1$; (iii) for all other $v \in X_i, f(v) = g(v) = h(v) = 0$. Then

$$\Gamma_i(s) = \begin{cases} \max_{s' \in \mathcal{S}_a, s'' \in \mathcal{S}_b} \Gamma_a(s') \cdot \Gamma_b(s'') & \text{such that } \{s', s''\} \leq_c s, \\ 0 & \text{otherwise.} \end{cases}$$

As before, the correctness of the recurrence follows from the definition of consistency.

It is straightforward to calculate this in $(3q^2)^{|X_i|}$ time; we will further improve this to $(2q^2)^{|X_i|}$ time as follows.

Notice that if we fix c_1 and c_2 , then d_1, d_2, e_1, e_2 get fixed for consistent states. Also, given f, g , and h , consider the vertices $B(f) = \{v \in X_i \mid f(v) = 0\}$. Then for each vertex $v \in B(f), g(v) = h(v) = 0$. Now consider $X = X_i \setminus B(f)$. For consistent states, the following relations hold: (i) $g^{-1}(1) \uplus h^{-1}(1) = X$; (ii) $g^{-1}(0) \setminus B(f) = h^{-1}(1)$ and $g^{-1}(1) = h^{-1}(0) \setminus B(f)$. Thus, if we are given the function g , we can completely determine h when we are looking at consistent states. Now, fix a function f . We define functions $F_a, F_b: 2^{X_i} \rightarrow [0, 1]$ in the following way. For a subset $Z \subseteq X_i \setminus B(f)$, define a function $g: X_i \rightarrow \{0, 1\}$ such that for any $v \in Z, g(v) = 1$ and $g(v) = 0$ otherwise. Now, define $F_a(Z) = \Gamma_a(\mathbf{a}, c_1, c_2, g)$ and $F_b(Z) = \Gamma_b(\mathbf{b}, c_1, c_2, g)$.

Then,

$$\Gamma_i(\mathbf{i}, c_1, c_2, f) = \begin{cases} 1 & \text{if } F_a * F_b(X_i \setminus B(f)) = \\ & \max_{Z \subseteq X_i \setminus B(f)} \{F_a(Z) + F_b((X_i \setminus B(f)) \setminus Z)\} = 2, \\ 0 & \text{otherwise.} \end{cases}$$

The correctness of this recurrence is same as the correctness of the previous recurrence. Due to fast subset convolution over the max-sum semiring [6], this can be calculated in $(2q^2)^{|X_i|}$ time. \square

3.2. Running time lower bounds. In this section, we given the proof of Theorem 3.2 by describing lower bounds on algorithmic running times for the ONCF-COLORING and CNCF-COLORING problems parameterized by treewidth.

We start by providing a running time lower bound on 2-ONCF-COLORING under ETH claimed in Theorem 3.2.

LEMMA 3.4. *2-ONCF-COLORING parameterized by treewidth t cannot be solved in $2^{o(t)} n^{\mathcal{O}(1)}$ time under ETH.*

Note that a reduction from 3-NAE-SAT to 2-ONCF-COLORING was given in Theorem 2 of [11]. Combining this with the fact that 3-NAE-SAT where each instance variable appears in at most 4 clauses cannot be solved in $2^{o(m)} n^{\mathcal{O}(1)}$ time under ETH [7], the reduction from 3-NAE-SAT to 2-ONCF-COLORING in [11] proves the above lemma. In this paper, we provide an alternative reduction from 3-SAT that gives further structural insight into the ONCF-COLORING problem and introduces a useful gadget, which we will introduce next.

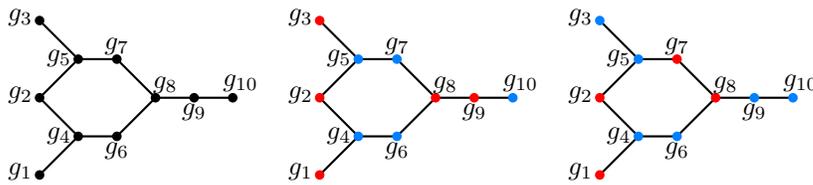


FIG. 1. The ONCF-gadget (left). Observe that if g_1, g_2 , and g_3 are all red, then g_9 must also be red (middle), and if one of g_1, g_2 , or g_3 is blue, then g_9 may be blue (right).

DEFINITION 3.5. An ONCF-gadget is a gadget on ten vertices, as depicted in Figure 1.

The objective of this gadget is the following. The vertices $\{g_1, g_2, g_3, g_{10}\}$ in Figure 1 will be the interaction points of the ONCF-gadget with the outside world. As will be proved in the following two lemmas, the gadget is designed so as to (i) disallow certain 2-ONCF-colorings and (ii) allow certain 2-ONCF-colorings on its interaction points.

LEMMA 3.6. Let G be a ONCF-gadget with a coloring $c: V(G) \rightarrow \{\text{red}, \text{blue}\}$ such that for all $4 \leq i \leq 9$ the neighborhood of g_i is ONCF-colored by c . If $c(g_1) = c(g_2) = c(g_3) = \text{red}$, then $c(g_9) = \text{red}$.

Proof. Suppose $c(g_1) = c(g_2) = c(g_3) = \text{red}$. Since $N(g_4) = \{g_1, g_2, g_6\}$, this implies $c(g_6) = \text{blue}$. Similarly, we find $c(g_7) = \text{blue}$. Since $N(g_8) = \{g_6, g_7, g_9\}$ now has two blue vertices, we conclude that $c(g_9) = \text{red}$. \square

LEMMA 3.7. Let G be a ONCF-gadget. Let $c': \{g_1, g_2, g_3\} \rightarrow \{\text{red}, \text{blue}\}$ be a partial 2-ONCF-coloring of G . If there exists $i \in [3]$ such that $c'(g_i) = \text{blue}$, then c' can be extended to a coloring c satisfying

1. for every $4 \leq i \leq 9$, the neighborhood of vertex g_i is ONCF-colored by c (contains at most one red, or at most one blue vertex); and
2. $c(g_9) = \text{blue}$, $c(g_8) = \text{red}$, $c(g_4) = c(g_5) = \text{blue}$, and $c(g_{10}) = \text{blue}$.

Proof. Let c equal c' on vertices g_1, g_2 , and g_3 and define $c(g_9) := \text{blue}$, $c(g_8) := \text{red}$, $c(g_4) := c(g_5) := \text{blue}$, and $c(g_{10}) := \text{blue}$. If $c'(g_1) = \text{blue}$ or $c'(g_2) = \text{blue}$, define $c(g_6) := \text{red}$ else define $c(g_6) := \text{blue}$. If $c'(g_2) = \text{blue}$ or $c'(g_3) = \text{blue}$, define $c(g_7) := \text{red}$, otherwise let $c(g_7) := \text{blue}$. This completes the definition of c . It is easy to verify that both requirements are satisfied by this coloring; refer to Figure 1 for an example coloring. \square

Now that we have introduced the necessary gadgets, we can prove the running time lower bound for 2-ONCF-COLORING using a reduction from 3-SAT.

Proof of Lemma 3.4. We show this by giving a reduction from 3-SAT. Given an instance of 3-SAT with variables x_1, \dots, x_n and clauses C_1, \dots, C_m , create a graph G as follows. Start by creating palette vertices R, R' , and B , and edges $\{R, R'\}$ and $\{R', B\}$. For each variable $i \in [n]$, create vertices u_i, v_i, w_i and add edges $\{u_i, v_i\}$ and $\{v_i, w_i\}$. For the remainder of the construction we will reuse the ONCF-gadget as defined in Definition 3.5. For each $j \in [m]$, add an ONCF-gadget G_j and connect g_{10} of this gadget to R . Add vertices s_j^1, s_j^2 , and s_j^3 and connect s_j^b to g_b in G_j for $b \in [3]$. Let clause $C_j := (\ell_1, \ell_2, \ell_3)$. Now if $\ell_b = x_i$ for some $i \in [n], b \in [3]$, connect s_j^b to u_i . Similarly, if $\ell_b = \neg x_i$, connect s_j^b to w_i . This concludes the construction of G , it remains to show that G is 2-ONCF-colorable if and only if the formula was satisfiable.

Suppose the satisfiability instance has satisfying assignment $\tau: \{x_1, \dots, x_n\} \rightarrow \{0, 1\}$, we show how to color G . Let $c(R) := c(R') := \text{red}$, and $c(B) := \text{blue}$. Let $c(v_i) := \text{blue}$ for all $i \in [n]$ and define $c(s_j^b) := \text{red}$ for all $b \in [3], j \in [m]$. Finally, if $\tau(x_i) = 1$, let $c(u_i) := \text{red}$ and $c(w_i) := \text{blue}$. Otherwise, let $c(u_i) := \text{blue}$ and $c(w_i) := \text{red}$. For each gadget G_m , vertex g_b for $b \in [3]$ has neighbor s_j^b . Let $v \in \{u_i, w_i \mid i \in [n]\}$ be the other neighbor of vertex s_j^b . Define $c(g_b)$ such that $c(g_b) \neq c(v)$. Since the formula was satisfied by τ , for each $j \in [m]$ there hereby exists $b \in [3]$ such that $c(g_b) = \text{blue}$. We use Lemma 3.7 to extend the partial coloring to color gadget G_m , with $c(g_{10}) = \text{blue}$ and $c(g_4) = c(g_5) = \text{blue}$. It is straightforward to verify that c is a 2-ONCF-coloring of G .

Suppose G has a 2-ONCF-coloring, we give a satisfying assignment τ . Assume without loss of generality that $c(R) := \text{red}$. Since $N(v_i) := \{w_i, u_i\}$ for all $i \in [n]$, it follows that $c(u_i) \neq c(w_i)$. We therefore define $\tau(x_i) := 1$ if $c(u_i) := \text{red}$ and $\tau(x_i) = 0$ if $c(w_i) := \text{red}$. Let C_j be a clause; we will show that τ satisfies C_j to conclude the proof. Suppose for contradiction that τ does not satisfy C_j . Then every vertex s_j^b for $b \in [3]$ had one neighbor in $\{u_i, w_i \mid i \in [n]\}$ that is blue in G . Thereby, its only other neighbor g_b in gadget G_j must be colored red. It follows from Lemma 3.6 that $c(g_9) := \text{red}$. Observe however that $N(g_{10}) := \{g_9, R\}$ and that both these vertices are red, contradicting that c is a 2-ONCF-coloring of G . Thus, the formula is satisfied by τ .

Note that the graph induced by $V(G) \setminus \{u_i, v_i, w_i \mid i \in [n]\}$ is a disjoint union of ONCF-gadgets and has treewidth two. As such, G has treewidth at most $3n + 2$.

In this reduction a 3-SAT formula ϕ on n variables and m clauses is reduced to a graph G with treewidth at most $3n + 2$. We proved that ϕ is satisfiable if and only if G has a 2-ONCF-coloring. Since 3-SAT cannot be solved in $2^{o(n)}n^{O(1)}$ time under ETH, this also implies that 2-CNCF-COLORING parameterized by treewidth t cannot be solved in $2^{o(t)}n^{O(1)}$ time under ETH. \square

LEMMA 3.8. *For $q \geq 3$, q -ONCF-COLORING parameterized by treewidth t cannot be solved in $(q - \varepsilon)^t n^{O(1)}$ time under SETH.*

Proof. It was shown in [17] that for a constant $q \geq 3$, q -COLORING cannot be solved in $(q - \varepsilon)^t n^{O(1)}$ time under SETH. For a graph G , let G' be the graph obtained by subdividing every edge of $E(G)$ once. It was shown in Theorem 3 of [11], that G has a q -coloring if and only if G' has a q -ONCF-coloring. Also, note that $tw(G') \leq tw(G)$ since it is a subdivision of G . Thus, for a constant $q \geq 3$, the lower bound of $(q - \varepsilon)^t n^{O(1)}$ on the running time of any algorithm under SETH follows. \square

LEMMA 3.9. *2-CNCF-COLORING parameterized by treewidth t cannot be solved in $2^{o(t)}n^{O(1)}$ time under ETH.*

Proof. In [11], a reduction of 2-CNCF-COLORING was given from 3-SAT. In this reduction a 3-SAT formula ϕ on n variables and m clauses is reduced to a graph G with treewidth at most $20m$. It was shown that ϕ is satisfiable if and only if G has a 2-CNCF-coloring. Since 3-SAT cannot be solved in $2^{o(m)}n^{O(1)}$ time under ETH, this also implies that 2-CNCF-COLORING parameterized by treewidth t cannot be solved in $2^{o(t)}n^{O(1)}$ time under ETH. \square

LEMMA 3.10. *For $q \geq 3$, q -CNCF-COLORING parameterized by treewidth t cannot be solved in $(q - \varepsilon)^t n^{O(1)}$ time under SETH.*

Proof. It was shown in [17] that for a constant $q \geq 3$, q -COLORING cannot be solved in $(q - \varepsilon)^t n^{O(1)}$ time under SETH. For a graph G , Theorem 3.1 of [1] constructs

a graph G' such that G has a q -coloring if and only if G' has a q -CNCF-coloring. The construction of G' requires the graphs G_k as described in the preliminaries (refer to Definition 2.1); these graphs were first constructed in [1].

Returning to the construction of G' , we obtain G' from G in the following manner: (i) for each vertex $v \in V(G)$ we add two copies G_q^{1v} and G_q^{2v} of G_q and make v adjacent to all vertices of G_q^{1v} and G_q^{2v} ; (ii) for each edge $e = \{u, v\} \in E(G)$ we add two copies G_{q-1}^{1e} and G_{q-1}^{2e} of G_{q-1} and make the vertices u and v adjacent to all vertices of G_{q-1}^{1e} and G_{q-1}^{2e} . This completes the construction of G' . For the completion of our proof it remains to show that $\text{tw}(G') \leq \text{tw}(G)$ in order to obtain a lower bound of $(q - \varepsilon)^t n^{\mathcal{O}(1)}$ on the running time of any algorithm under SETH.

CLAIM 3.11. *For a graph G_k , $\text{tw}(G_k) \leq k - 1$.*

Proof. We prove our statement by induction on k . In the base case, it is true that $\text{tw}(G_1) = 0$ and $\text{tw}(G_2) = 1$. Let the induction hypothesis be that for any $k' < k$, $\text{tw}(G_{k'}) \leq k' - 1$. We prove the statement for G_k . By construction, G_k contains a clique C on k vertices. We create a bag X with all the vertices of C . By the induction hypothesis, for each copy of G_{k-1} we have a tree decomposition \mathcal{T}_{k-1} of width $k - 2$. Similarly, let \mathcal{T}_{k-2} be a tree decomposition of G_{k-2} with width $k - 3$. Note that by construction, each copy of G_{k-1} only has edges with a single vertex, say v from the clique C . To each bag of the corresponding tree decomposition, we add the vertex v , thereby making the treewidth of the tree decomposition at most $k - 1$. We pick an arbitrary bag of the tree decomposition and attach it to the bag X containing the vertices of C . Similarly, each copy of G_{k-2} only has edges with the end points of a single edge, say $\{u, v\}$ from the clique C . To each bag of the corresponding tree decomposition, we add the vertices u, v , thereby making the treewidth of the tree decomposition at most $k - 1$. We pick an arbitrary bag of the tree decomposition and attach it to the bag X . The resulting tree decomposition has width at most $k - 1$. Thus, $\text{tw}(G_k) \leq k - 1$. \lrcorner

This helps us to show the desired treewidth bound for G' .

CLAIM 3.12. *For a graph G' , $\text{tw}(G') \leq \max\{\text{tw}(G), q\}$.*

Proof. The construction of a desired tree decomposition is similar to the construction given in the previous claim. Let \mathcal{T} be a tree decomposition of G . By construction, each copy of G_{q-1} , that is added to G to form G' , is attached to a single vertex in $V(G)$, say v . From the previous claim, we have a tree decomposition \mathcal{T}_{q-1} of width $q - 2$ for this copy of G_{q-1} . To each bag of \mathcal{T}_{q-1} , we add the vertex v , thereby increasing the treewidth to at most $k - 1$. We pick an arbitrary bag of \mathcal{T} that contains v and an arbitrary bag of \mathcal{T}_{q-1} and attach them together. Similarly, each copy of G_{q-2} , that is added to G to form G' , is attached to the end points of a single edge in $E(G)$, say $\{u, v\}$. From the previous claim, we have a tree decomposition \mathcal{T}_{q-2} of width $q - 3$ for this copy of G_{q-2} . To each bag of \mathcal{T}_{q-2} , we add the vertices u, v , thereby increasing the treewidth to at most $k - 1$. We pick an arbitrary bag of \mathcal{T} that contains the edge $\{u, v\}$ and an arbitrary bag of \mathcal{T}_{q-2} and attach them together. Note that the resulting tree decomposition is a tree decomposition of G' and has width at most $\max\{\text{tw}(G), q\}$. Thus, we are done. \lrcorner

As q is a constant, we can restrict ourselves to the case that $\text{tw}(G) \geq q$. Now, $\text{tw}(G') \leq \max\{\text{tw}(G), q\} \leq \text{tw}(G)$. Thus, for a constant $q \geq 3$, the lower bound of $(q - \varepsilon)^t n^{\mathcal{O}(1)}$ on the running time of any algorithm under SETH follows. \square

Thus, using Lemmas 3.4, 3.8, 3.10, and 3.9 we complete the proof of Theorem 3.2.

4. Kernelization. In this section, we will study the kernelizability of the ONCF- and CNCF-coloring problems, when parameterized by the size of a vertex cover. We prove the following two theorems to obtain a dichotomy on the kernelization question.

THEOREM 4.1. *q -ONCF-COLORING for $q \geq 2$ and q -CNCF-COLORING for $q \geq 3$, parameterized by vertex cover size do not have polynomial kernels, unless $\text{NP} \subseteq \text{coNP/poly}$.*

Sections 4.1 and 4.2 together give a full proof of Theorem 4.1.

THEOREM 4.2. *2-CNCF-COLORING parameterized by vertex cover size k has a generalized kernel of size $\mathcal{O}(k^9)$.*

We prove the above theorem in section 4.3. Note that by using an NP-completeness reduction, this results in a polynomial kernel for 2-CNCF-COLORING parameterized by vertex cover size. We also obtain an $\mathcal{O}(k^2 \log k)$ kernel for an extension problem of 2-CNCF-COLORING and this is described in section 4.4.

4.1. Kernel lower bounds for q -ONCF-COLORING. In this part, we begin the proof of Theorem 4.1 by showing that q -ONCF-COLORING parameterized by vertex cover size has no polynomial kernel when q is at least 2. We first show the relevant bound for $q = 2$ and then use a polynomial parameter transformation to obtain the general lower bound.

For the construction in the following proof, we will again use the ONCF-gadget that was introduced in Definition 3.5 (and shown in Figure 1). Recall the relevant properties of this gadget that were given in Lemmas 3.6 and 3.7.

LEMMA 4.3. *2-ONCF-COLORING parameterized by vertex cover size does not have a polynomial kernel, unless $\text{NP} \subseteq \text{coNP/poly}$.*

Proof. We show this by giving an OR-cross-composition (see Definition 2.9) from CLIQUE to 2-ONCF-COLORING parameterized by vertex cover size. It is well known that CLIQUE is an NP-hard problem [10]. Therefore from Theorem 2.10, an OR-cross-composition from CLIQUE to 2-ONCF-COLORING parameterized by vertex cover size implies that the latter does not have a polynomial kernel unless $\text{NP} \subseteq \text{coNP/poly}$.

We proceed with the description of the OR-cross-composition. For the sake of brevity, in this proof, we use 2-ONCF-coloring and ONCF-coloring interchangeably. We define a polynomial equivalence relation \mathcal{R} (see Definition 2.8) as follows. Let 2 instances of CLIQUE be equivalent under \mathcal{R} if the graphs have the same number of vertices and they ask for a clique of the same size. It is easy to verify that \mathcal{R} is a polynomial equivalence relation. Suppose we are given t instances of clique that are equivalent under \mathcal{R} , label them as X_1, \dots, X_t . Let every instance have n vertices and ask for a clique of size k , and enumerate the vertices in each instance arbitrarily. We create an instance G for 2-ONCF-coloring by the following steps (see Figure 2 for a sketch of G).

1. Create vertices $R, R', B,$ and B' . Connect R to R' , R' to B , and B to B' . This ensures that R and B receive distinct colors in any coloring. (We will without loss of generality assume that R receives the color red, and B receives the color blue). Thus, the vertices R and B can be thought of as the palette for any 2-ONCF-coloring for G .
2. Create $3t$ vertices $\{y_\ell, y'_\ell, y''_\ell \mid \ell \in [t]\}$ and let Y be the set containing these vertices. These vertices will be used to “select” which instance has a clique of size k .
3. Add a vertex a and connect a to all vertices in Y . Add vertices $a_1, a'_1, a_2,$ and a'_2 and edges $\{a, a_1\}, \{a, a_2\}, \{a_1, a'_1\},$ and $\{a_2, a'_2\}$. Finally, connect a'_1

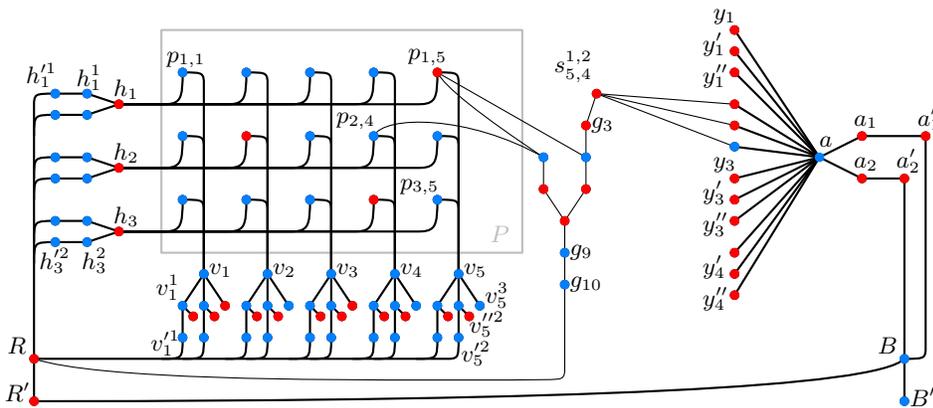


FIG. 2. A sketch of the constructed graph G for $k = 3$, $n = 5$, and $t = 4$, assuming edge $\{5, 4\}$ is missing in instance X_2 and present in all other instances. All vertices created in steps 7 and 8 of the construction are omitted for simplicity, except for vertex $s_{5,4}^{1,2}$ and gadget $G_{5,4}^{1,2}$.

to B and connect a_2' to B . This ensures that vertices a_1 and a_2 are red in any valid ONCF-coloring. Thereby, a must have exactly one blue neighbor, implying exactly one vertex in Y is blue. The vertex that is colored blue in Y will then correspond to the input instance that has a clique of size k .

4. Add vertices $p_{i,j}$ for $i \in [k]$, $j \in [n]$. Let P be the set consisting of these vertices. These vertices will be used to select the vertices that correspond to a clique in one of the input instances.
5. Add a vertex h_i for all $i \in [k]$. Connect h_i to $p_{i,j}$ for all $j \in [n]$. For each vertex h_i , add vertices h_i^1, h_i^2, h_i^3 , and h_i^4 . Connect h_i to h_i^1 and h_i^2 . Connect h_i^1 to h_i^3 , connect h_i^2 to h_i^4 . Connect h_i^3 and h_i^4 to R , in order to ensure that h_i^3 and h_i^4 will both be colored blue in any ONCF-coloring. Let H be the set of all vertices created in this step.
6. Add a vertex v_j for $j \in [n]$. Connect v_j to $p_{i,j}$ for all $i \in [k]$. Create vertex v_j^3 and connect it to v_j . Furthermore, create vertices $v_j^x, v_j^{x'}$, and $v_j^{x''}$ for all $x \in [2]$ and connect v_j^x to v_j . Furthermore, connect $v_j^{x'}$ to v_j^x . Then, connect v_j^x to $v_j^{x'}$ and $v_j^{x'}$ to R for all $x \in [2]$. By this construction, vertex v_j has at least two blue neighbors, and one neighbor whose coloring can be freely chosen. Let V be the set of all vertices created in this step.
7. For every $i, i' \in [k]$ and $j, j' \in [n]$, we add a vertex $s_{j,j'}^{i,i'}$, and let S be the set containing all these vertices. For each $\ell \in [t]$, $i, i' \in [k]$, and $j, j' \in [n]$, add the edges $\{s_{j,j'}^{i,i'}, y_\ell\}$, $\{s_{j,j'}^{i,i'}, y'_\ell\}$, and $\{s_{j,j'}^{i,i'}, y''_\ell\}$ if and only if $\{j, j'\} \notin E(X_\ell)$. The idea is that vertex $s_{j,j'}^{i,i'}$ verifies that if we select vertices j and j' to be part of the clique, then no instance X_ℓ where $\{j, j'\} \notin E(X_\ell)$ can be selected as the yes-instance. To do this, we add additional gadgets in the following step.
8. For each $i, i' \in [k]$ and $j, j' \in [n]$ add a new ONCF-gadget $G_{j,j'}^{i,i'}$. Identify vertex g_1 of the gadget with $p_{i,j}$ and identify vertex g_2 of the gadget with $p_{i',j'}$. Add the edge $\{g_3, s_{j,j'}^{i,i'}\}$. Finally, connect vertex g_{10} to R .

In the remainder, we observe that $c(R) \neq c(B)$ for any 2-ONCF-coloring of G . Thereby, we can safely rename the colors such that $c(R) = \text{red}$ and $c(B) = \text{blue}$.

CLAIM 4.4. *Let c be any 2-ONCF-coloring of G , then $c(a_1) = c(a_2) = \text{red}$, $c(v_j^1) = c(v_j^2) = \text{blue}$ for all $j \in [n]$, and $c(h_i^1) = c(h_i^2) = \text{blue}$ for all $i \in [k]$.*

Proof. This follows immediately from the fact that there is a degree-2 vertex connecting these vertices to R or B , respectively. \lrcorner

CLAIM 4.5. *For any 2-ONCF-coloring c of G , there exists exactly one vertex $y^* \in Y$ such that $c(y^*) = \text{blue}$ and for all other vertices $y \in Y \setminus \{y^*\}$, $c(y) = \text{red}$.*

Proof. Observe that $N(a) = \{a_1, a_2\} \cup Y$ and that $c(a_1) = c(a_2) = \text{red}$ by Claim 4.4. Thereby, a must have a unique blue neighbor and this neighbor is in Y . \lrcorner

CLAIM 4.6. *Let $G_{j,j'}^{i,i'}$ be a gadget and let c be a 2-ONCF-coloring of G . Then $c(g_1) = c(g_2) = \text{red}$ in this gadget implies that $c(g_3) = \text{blue}$.*

Proof. Suppose $c(g_1) = c(g_2) = c(g_3) = \text{red}$. It follows from Lemma 3.6 that thereby $c(g_9) := \text{red}$. Observe however that $N(g_{10}) = \{R, g_9\}$ by definition. Since both these vertices are red, this contradicts the assumption that c is a proper ONCF-coloring. \lrcorner

CLAIM 4.7. *For any 2-ONCF-coloring c of G , there exist distinct j_1, \dots, j_k such that $c(p_{i,j_i}) = \text{red}$ and for all other $p \in P$, $c(p) := \text{blue}$.*

Proof. We start by showing that for each $i \in [k]$, there is exactly one $j \in [n]$ such that $c(p_{i,j}) = \text{red}$. Consider the neighborhood of vertex h_i . $N(h_i) := \{h_i^1, h_i^2\} \cup \{p_{i,j} \mid j \in [n]\}$. Since $c(h_i^1) = c(h_i^2) = \text{blue}$ by Claim 4.4, it follows that indeed $\{h_i^1, h_i^2\} \cup \{p_{i,j} \mid j \in [n]\}$ contains exactly one red vertex, let this be vertex p_{i,j_i} . It remains to show that all j_i are distinct.

We show this by proving that for each $j \in [n]$, there is at most one $i \in [k]$ such that $c(p_{i,j}) = \text{red}$. Consider vertex v_j , observe that $N(v_j) := \{v_j^1, v_j^2, v_j^3\} \cup \{p_{i,j} \mid i \in [k]\}$. Since $c(v_j^1) = c(v_j^2) = \text{blue}$ by Claim 4.4, it follows that v_j has a unique red neighbor, and thus there is at most one $i \in [k]$ such that $c(p_{i,j}) = \text{red}$. Hereby, the claim follows. \lrcorner

Using the claims above, we can now prove the correctness of the given cross-composition.

CLAIM 4.8. *If there exists an instance X_ℓ that has a clique of size k , then G can be 2-ONCF-colored.*

Proof. Let ℓ be such that X_ℓ is a yes-instance for clique. Choose $j_1, \dots, j_k \in [n]$ such that these vertices form a clique in X_ℓ . We now give an ONCF-coloring c for G ; see Figure 2 for an example ONCF-coloring of G .

1. Let $c(R) := c(R') := \text{red}$ and $c(B) := c(B') := \text{blue}$.
2. Let $c(y_\ell) := \text{blue}$. For all other vertices in $y \in Y$, let $c(y) := \text{red}$.
3. Let $c(a) := \text{blue}$ and $c(a_1) := c(a_2) := c(a'_1) := c(a'_2) := \text{red}$.
4. Let $c(p_{i,j_i}) := \text{red}$ for all $i \in [k]$. For all other vertices $p \in P$, let $c(p) := \text{blue}$.
5. Let $c(h_i) := \text{red}$ for all $i \in [k]$. Let $c(h) := \text{blue}$ for all other vertices $h \in H$.
6. For $j \in [n]$, let $c(v_j^3) := \text{red}$ if $j \notin \{j_1, \dots, j_k\}$, let $c(v_j^3) := \text{blue}$ otherwise. Let $c(v_j^{''x}) := \text{red}$ for all $x \in [2]$. Let $c(v) := \text{blue}$ for all remaining vertices $v \in V$.
7. Let $c(s) := \text{red}$ for all $s \in S$.
8. It remains to color the introduced gadgets. Observe that vertices g_1 and g_2 of each gadget have already been colored, as they were identified with vertices from P . We now proceed as follows. Define $c(g_3) := \text{blue}$ whenever $s_{j,j'}^{i,i'}$ has no blue neighbor in Y and define $c(g_3) := \text{red}$ otherwise. Observe that since j_1, \dots, j_k form a clique in instance X_ℓ , it never happens that $c(g_1) = c(g_2) = c(g_3) = \text{red}$ by this definition. Color the remainder of each gadget

using Lemma 3.7, such that the coloring satisfies property 2 of the lemma statement.

This defines a 2-coloring of G ; it remains to verify that c is indeed a 2-ONCF-coloring. We consider the neighborhood of each vertex in G .

1. $N(R) := \{R'\} \cup \{v_j^1, v_j^2 \mid j \in [n]\} \cup \{h_i^1, h_i^2 \mid i \in [n]\}$. Since $c(R') = \text{red}$ and $c(x) = \text{blue}$ for all $x \in \{v_j^1, v_j^2 \mid j \in [n]\} \cup \{h_i^1, h_i^2 \mid i \in [n]\}$, $N(R)$ is ONCF-colored. $N(B) := \{R', B', a_1', a_2'\}$, of which only R' is red. Thus, $N(B)$ is ONCF-colored. Furthermore, $|N(B')| = 1$ and thereby it is trivially ONCF-colored, and $N(R') = \{R, B\}$ which have distinct colors as desired.
2. For any vertex $y \in Y$, $N(Y)$ contains vertex a which is colored blue. Furthermore, $N(Y) \setminus \{a\} \subseteq S$ and all vertices in S are red.
3. $N(a) = Y \cup \{a_1, a_2\}$. Y contains exactly one blue vertex, and a_1 and a_2 are colored red. $N(a_1) := \{a, a_1'\}$, which have distinct colors as desired. Similarly, $N(a_2) := \{a, a_2'\}$ and these vertices are blue and red, respectively. Finally, $N(a_1') := \{a_1, B\}$ and $N(a_2') := \{a_2, B\}$, it is easy to verify that these are ONCF-colored.
4. For $i \in [k]$ and $j \in [n]$, $N(p_{i,j})$ contains vertex h_j which is red. Furthermore, $N(p_{i,j}) \setminus \{h_j\}$ only contains vertices from V , which are blue, and vertices g_4 and g_5 from numerous ONCF-gadgets, which are also blue. Thereby it has red as a unique color in its neighborhood.
5. For $i \in [k]$, $N(h_i) := \{p_{i,j} \mid j \in [n]\} \cup \{h_i^1, h_i^2\}$. Observe that all vertices in $N(h_i)$ are blue, except vertex p_{i,j_i} which is red. For $x \in [2]$, $N(h_i^x) := \{h_i^x, h_i\}$ and these vertices receive distinct colors. $N(h_i^x) := \{R, h_i^x\}$ and these vertices also receive distinct colors.
6. For $j \in [n]$, we observe that v_j has exactly one red neighbor in P and all its other neighbors are blue. Vertices v_j^1 and v_j^2 both have exactly one red neighbor, namely, vertex $v_j'^1$ or $v_j'^2$, respectively. Vertices v_j^x have one blue and one red neighbor for $x \in [2]$. The vertices v_j^x for $x \in [2]$ and vertex v_j^3 have degree one and are thus ONCF-colored by definition.
7. For $i, i' \in [k]$, $j, j' \in [n]$, vertex $s_{j,j'}^{i,i'}$ has neighbors in Y and vertex g_3 in gadget $G_{j,j'}^{i,i'}$. It follows from the definition of the coloring of g_3 and the fact that Y has at most one blue vertex that $s_{j,j'}^{i,i'}$ has exactly one blue neighbor.
8. It remains to verify that all gadget vertices are ONCF-colored properly; consider the vertices of gadget $G_{j,j'}^{i,i'}$. The neighborhoods of vertices g_4, g_5, \dots, g_9 are ONCF-colored by definition. Vertices g_1 and g_2 were identified with vertices from P and have already been discussed above. $N(g_3) = \{g_5, s_{j,j'}^{i,i'}\}$ and these are blue and red, respectively. $N(g_{10}) = \{g_9, R\}$ and these are also blue and red. ┘

CLAIM 4.9. *If G can be 2-ONCF-colored, then there exists $\ell \in [t]$ such that instance X_ℓ has a clique of size k .*

Proof. Let a 2-ONCF-coloring c of G be given. By Claim 4.5, there exists $y \in Y$ with $c(y) := \text{blue}$. Pick ℓ such that $y \in \{y_\ell, y'_\ell, y''_\ell\}$. By Claim 4.7, take distinct $j_1, \dots, j_k \in [n]$ such that $c(p(i, j_i)) = \text{red}$. We will show that instance X_ℓ has a clique of size k , by proving that vertices j_1, \dots, j_k form a clique in X_ℓ .

Suppose not, then there exist $i, i' \in [k]$ such that j_i and $j_{i'}$ are not connected by an edge in X_ℓ . We show that this leads to a contradiction. Consider gadget $G_{j_i, j_{i'}}^{i, i'}$. Vertices g_1 and g_2 of this gadget are colored red, as they were identified with vertices

p_{i,j_i} and $p_{i',j_{i'}}$, respectively. It follows from Claim 4.6 that thereby $c(g_3) := blue$ in this gadget. Now consider the neighborhood of vertex $s_{j_i,j_{i'}}^{i,i'}$. It contains vertex g_3 from gadget $G_{j_i,j_{i'}}^{i,i'}$ and the vertices $y_\ell, y'_\ell, y''_\ell$ since edge $\{j_i, j_{i'}\}$ does not occur in instance X_ℓ . It now follows that vertex $s_{j_i,j_{i'}}^{i,i'}$ has at least two blue and two red neighbors in G , which contradicts that c is an ONCF-coloring of G . \lrcorner

From Claims 4.8 and 4.9, it follows that G can be 2-ONCF-colored if and only if there exists an ℓ such that X_ℓ has a clique of size k . To prove the lower bound, it remains to bound the size of a vertex cover in G . Since Y is an independent set in G , it follows that $V(G) \setminus Y$ is a vertex cover of G . Observe that

$$|V(G) \setminus Y| = \mathcal{O}(n^2 \cdot k^2).$$

To conclude, since we have given a cross-composition from CLIQUE to 2-ONCF-COLORING parameterized by vertex cover size, the lower bound now follows from Theorem 2.10. \square

Now, we use the lower bound obtained for 2-ONCF-COLORING in Lemma 4.3 and exhibit a polynomial parameter transformation to obtain the general lower bound for q -ONCF-COLORING for all $q \geq 2$. This completes the lower bound results for q -ONCF-COLORING claimed in Theorem 4.1.

LEMMA 4.10. *For any $q \geq 2$, q -ONCF-COLORING parameterized by vertex cover size does not have a polynomial kernel, unless $\text{NP} \subseteq \text{coNP/poly}$.*

Proof. We prove the result by giving a polynomial parameter transformation from 2-ONCF-COLORING parameterized by vertex cover size to q -ONCF-COLORING parameterized by vertex cover size for any constant $q > 2$. By Theorem 2.7 and Lemma 4.3, this implies that q -ONCF-COLORING parameterized by vertex cover size, does not have a polynomial kernel unless $\text{NP} \subseteq \text{coNP/poly}$ for $q \geq 2$. We will do this by adding additional structures to the graph, that ensure that the original graph is colored using only 2 colors, and that for any vertex in the original graph, its ONCF-color is also one of these two colors. Suppose we are given a graph G for 2-ONCF-COLORING, we show how to obtain G' for q -ONCF-COLORING.

1. Start by initiating G' as G . Let $V := V(G)$.
2. Add $2q$ vertices $x_1^0, x_2^0, \dots, x_q^0$ and $x_1^1, x_2^1, \dots, x_q^1$. Let X be the set of all these vertices. Add a clique on $x_1^0, x_2^0, \dots, x_q^0$. Connect x_j^1 to x_i^0 for all $i \neq j$ with $i, j \in [q]$. Finally, subdivide all the edges between vertices in X . (Thus, vertices $x_1^0, x_2^0, \dots, x_q^0$ form a subdivided clique in G'). Let the set of vertices used to subdivide these edges be X' .
3. Add $2(q - 2)$ vertices y_ℓ^0, y_ℓ^1 for $\ell \in [q - 2]$; let Y be the set containing all these vertices. Connect y_ℓ^0 and y_ℓ^1 to all vertices in $\{x_i^0, x_i^1 \mid i \in [q] \wedge i \neq \ell\}$. Then, connect y_ℓ^0 to x_ℓ^0 and connect y_ℓ^1 to x_ℓ^1 . Finally, connect y_ℓ^0 and y_ℓ^1 to every vertex $v \in V$.
4. For $b \in \{0, 1\}$ and $\ell \in [q - 2]$, use a subdivided edge to connect y_ℓ^b to x_j^0 for all $j \neq \ell$. Let Y' be the set containing all vertices used for subdividing these edges. This ensures that y_ℓ^b always receives color ℓ .

CLAIM 4.11. *If $\chi_{\text{ON}}(G') \leq q$, then $\chi_{\text{ON}}(G) \leq 2$.*

Proof. Suppose G' has a q -ONCF-coloring $c': V(G') \rightarrow [q]$, we will now show that G has a 2-ONCF-coloring. It is easy to observe that $c'(x_i^0) = c'(x_i^1)$ for all i and, furthermore, $c'(x_i^0) \neq c'(x_j^0)$ for all $i \neq j \in [q]$, by the subdivided edges

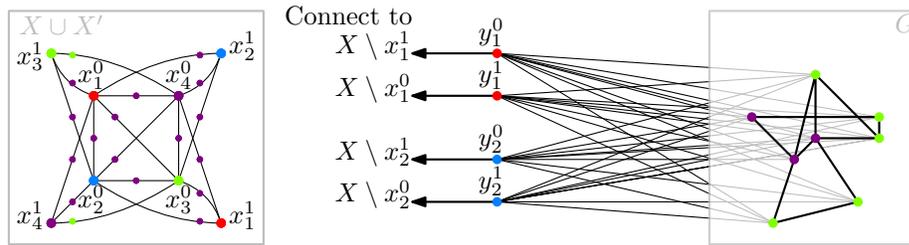


FIG. 3. Reduction from G to G' for $q = 5$. The subdivided edges from Y to X added in step 4 of the construction are omitted for simplicity.

introduced in step 2. By this observation, we may assume without loss of generality that $c'(x_i^0) = c'(x_i^1) = i$. It is easy to observe using step 4 of the construction, that in such a coloring $c'(y_\ell^0) = c'(y_\ell^1) = \ell$. Thereby, for every $\ell \in [q-2]$, $N(v)$ contains two vertices of color ℓ , for all $v \in V$. This implies that for any $\ell \in [q-2]$ and $v \in V$, we know that ℓ is not the color that ensures that $N(v)$ is q -ONCF-colored.

Furthermore, $N(y_\ell^0)$ contains two vertices of color i for all $i \neq \ell$ (namely, x_i^0 and x_i^1), and one vertex of color ℓ . Thereby, no vertex in v can have color ℓ . This implies that only two colors are used in V , namely, q and $q-1$. We conclude that the coloring c restricted to vertices in V is a 2-ONCF-coloring for G , after renaming the colors to $\{1, 2\}$. \lrcorner

CLAIM 4.12. *If $\chi_{\text{ON}}(G) \leq 2$, then $\chi_{\text{ON}}(G') \leq q$.*

Proof. Suppose G has a 2-ONCF-COLORING-coloring c , we show how to q -ONCF-color G' . First of all, let $c'(x_i^0) = c'(x_i^1) = i$ for all $i \in [q]$ and let $c'(y_\ell^0) = c'(y_\ell^1) = \ell$ for all $\ell \in [q-2]$. For $v \in V$, let $c'(v) = q-1$ when $c(v) = 1$ and let $c'(v) = q$ otherwise. For the vertex x^0 on the subdivided edge from x_q^0 to x_{q-1}^1 , let $c(x^0) = q-1$. Similarly, for the vertex x^1 on the subdivided edge from x_q^1 to x_{q-1}^0 , let $c(x^1) = q-1$. For all remaining vertices, let $c'(v) := q$. It remains to show that this gives a q -ONCF-coloring of G' . See Figure 3 for a sketch of G' .

1. The neighborhoods of vertices in V are trivially q -ONCF-colored since they have no vertices of color q or $q-1$ outside V , and c was a valid 2-ONCF-coloring.
2. For $i \in [q-2]$, $b \in \{0, 1\}$ vertex x_i^b has exactly one neighbor of color i , namely, y_i^b . For $i \in \{q-1, q\}$, vertices in x_i^b have exactly one neighbor of color $q-1$, namely, x^b or $x^{(1-b)}$. Vertices in X' have degree 2 and have neighborhoods that are q -ONCF-colored by this definition.
3. Vertices y_ℓ^0 and y_ℓ^1 for $\ell \in [q-2]$ each have exactly one neighbor of color ℓ and are thus q -ONCF-colored.
4. Neighborhoods of the vertices used for subdividing edges always have two vertices of distinct color, by definition. \lrcorner

Since G' is a copy of G to which we add $\mathcal{O}(q)$ additional vertices, it follows that the vertex cover of G' is bounded by $k + \mathcal{O}(q) = \mathcal{O}(k)$, where k is the size of a vertex cover in G . Thus, we have given a polynomial-parameter transformation from 2-ONCF-COLORING to q -ONCF-COLORING with both problems having vertex cover size as parameter, and the theorem statement follows from Theorem 2.7 and Lemma 4.3. \square

4.2. Kernel lower bound for CNCF-COLORING. In this part, we complete the proof of Theorem 4.1 by showing that q -CNCF-COLORING parameterized by

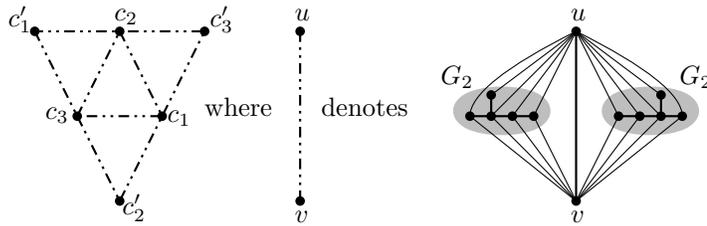


FIG. 4. A palette-gadget (left) where every dashed edge should be interpreted as the gadget depicted on the right.

vertex cover size has no polynomial kernel when q is at least 3. We will use the gadgets from Definition 2.1 to define our own palette-gadget C_q .

DEFINITION 4.13. To create a palette-gadget C_q , start from a complete graph on vertices c_1, \dots, c_q . Then, add vertices c'_i for $i \in [q]$ and connect c'_i to c_j for all $i \neq j$. Let $D := \{c_i, c'_i \mid i \in [q]\}$ be the set of distinguished vertices of the gadget. Finally, for each edge $\{u, v\} \in E(C_q)$ with $u, v \in D$, add two new distinct copies of G_{q-1} and connect all vertices in these copies to both u and v . See Figure 4 for an example of the palette-gadget C_3 .

The next two lemmas are used to establish that a palette gadget can indeed serve as a color palette for CNCF-Coloring.

LEMMA 4.14. Let G be a graph and let C be a set of vertices such that $G[C]$ is isomorphic to the palette-gadget C_q for some $q \geq 3$. Let f be a q -CNCF-coloring of G . Then $f(c_i) \neq f(c_j)$ for all $i, j \in [q]$ with $i \neq j$. Furthermore, $f(c_i) = f(c'_i)$ for all $i \in [q]$.

Proof. We show this by showing that if $\{u, v\}$ is an edge such that there are two distinct copies of G_{q-1} , say G_{q-1}^1 and G_{q-1}^2 , that are connected to both u and v and no other vertices in the graph, then $f(u) \neq f(v)$ in any q -CNCF-coloring of G . The results of the lemma statement then follow from the definition of the palette.

Suppose for contradiction that there exists a q -CNCF-coloring f with $f(u) = f(v)$. Let i be a color such that $|\{f(u) = i \mid u \in N[v]\}| = 1$. Observe that $i \neq f(u)$ as $f(u) = f(v)$ and $u, v \in N[v]$. Therefore, either G_{q-1}^1 or G_{q-1}^2 does not use color i , without loss of generality let this be G_{q-1}^1 . We show that thereby $\chi_{CN}^*(G_{q-1}^1) = q - 2$, which contradicts Lemma 2.2.

Define partial coloring f^* of G_{q-1}^1 as follows. For any $x \in V(G_{q-1}^1)$ with $f(x) \neq f(u)$, let $f^*(x) := f(x)$. For any $x \in V(G_{q-1}^1)$ with $f(x) = f(u)$, leave $f^*(x)$ undefined. Observe that hereby, the range of f^* is a subset of $[q] \setminus \{i, f(u)\}$ and thus f^* defines a $(q - 2)$ -coloring of G_{q-1}^1 . From the correctness of f and the fact that any vertex in G_{q-1}^1 has two neighbors of color $f(v) = f(u)$ under f , it follows that f^* is a partial $(q - 2)$ -CNCF-coloring of G_{q-1}^1 , which is a contradiction. \square

LEMMA 4.15. Let C_q be a palette-gadget for $q \geq 3$. Then there exists a q -CNCF-coloring $f: V(C_q) \rightarrow [q]$ such that

1. $f(c_i) = f(c'_i) = i$ for all $i \in [q]$, and
2. for all $i \in [q]$, $N[c_i]$ contains exactly one vertex of color i .

Proof. We start by defining $f(c_i) := f(c'_i) := i$ for all $i \in [q]$. Let $D := \{c_i, c'_i \mid i \in [q]\}$, observe that the color of all vertices in D has now been defined. All vertices in $V(C_q) \setminus D$ induce distinct copies of G_{q-1} . Consider an arbitrary copy of G_{q-1} in

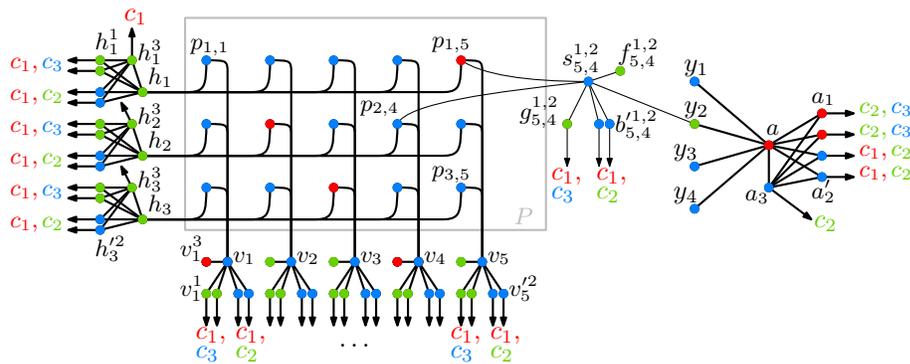


FIG. 5. A sketch of the constructed graph G for $q = 3$, $k = 3$, $n = 5$, and $t = 4$. All vertices created in steps 7 and 8 are omitted, except those created with $i = 1$, $i' = 2$, $j = 5$, and $j' = 4$, assuming edge $\{4, 5\}$ is not present in instance X_2 . The palette C is omitted; edges to the palette are drawn as arrows.

C_q , and suppose it was added to the palette gadget for the edge $\{d, d'\}$ with $d, d' \in D$. Color all vertices of this G_{q-1} with a color $x \in [q]$ such that $f(d) \neq x \neq f(d')$. Observe that such a color exists since $q \geq 3$.

Both requirements are satisfied by the definition of f ; it remains to show that f is a q -CNCF-coloring of C_q . Vertices c_i and c'_i have color i and have no neighbors of color i , and are thereby properly CNCF-colored. Vertices in $V(G) \setminus D$ are distinct copies of G_{q-1} . It is easy to verify that in $N(G_{q-1})$ there are two vertices with a unique color, corresponding to the edge for which it was added. Since these colors are not used to color G_{q-1} , the result follows. \square

Using the gadget introduced above, we now prove the kernelization lower bound.

LEMMA 4.16. *For any $q \geq 3$, q -CNCF-COLORING parameterized by vertex cover size does not have a polynomial kernel, unless $\text{NP} \subseteq \text{coNP/poly}$.*

Proof. To prove this theorem, we will give a cross-composition starting from CLIQUE that is very similar to the one given in Lemma 4.3.

We define the same polynomial equivalence relation; let two instances of CLIQUE be equivalent if the graphs have the same number of vertices and they ask for a clique of the same size. Suppose we are given t instances of clique that are equivalent under this relation, labeled X_1, \dots, X_t . Let every instance have n vertices and ask for a clique of size k . We enumerate the vertices in each instance arbitrarily as $1, \dots, n$. We create an instance G for q -CNCF-COLORING by the following steps, refer to Figure 5 for a sketch of G .

1. Create a palette-gadget C_q with distinguished vertices c_1, \dots, c_q and c'_1, \dots, c'_q by Definition 4.13.
2. Create t vertices y_1, \dots, y_t and let $Y := \{y_\ell \mid \ell \in [t]\}$. The idea is that exactly one of these vertices y_ℓ will receive color 2 in any q -CNCF-coloring, and this indicates that X_ℓ is a yes-instance for clique.
3. Add a vertex a and connect a to y_ℓ for all $\ell \in [t]$. Create vertices a_1 and a'_1 and connect both of these to vertices c_2 and c_3 in the palette-gadget. Furthermore, create vertices a_2 and a'_2 and connect them to vertices c_1 and c_2 . Finally, create vertex a_3 , connect a_3 to c_2 , a_1 , a'_1 , a_2 , and a'_2 . Connect a to a_1 , a'_1 , a_2 , a'_2 , and a_3 . The idea is that in any q -CNCF-coloring, vertices a_1 and a'_1 receive the color of c_1 , a_2 and a'_2 receive the color of c_3 and that

- a_3 has exactly one neighbor of color 2 and two neighbors of both remaining colors, implying that the color of a cannot be 2.
4. Add vertices $p_{i,j}$ for all $i \in [k], j \in [n]$. Let P be the set containing all of these vertices. The idea is that vertices in P receive colors 1 and 3, such that for every i there is exactly one vertex $p_{i,j}$ of color 1. The vertices of color 1 will correspond to the vertices that form a clique in one of the input instances.
 5. Add a vertex h_i for all $i \in [k]$ and connect h_i to $p_{i,j}$ for all $j \in [n]$. Add vertices $h_i^1, h_i^{1'}, h_i^2, h_i^{2'},$ and h_i^3 . Add edges $\{c_1, h_i^1\}, \{c_1, h_i^{1'}\}, \{c_3, h_i^1\}, \{c_3, h_i^{1'}\}, \{c_1, h_i^2\}, \{c_1, h_i^{2'}\}, \{c_2, h_i^2\},$ and $\{c_2, h_i^{2'}\}$. For all $i \in [k]$, add a vertex h_i^3 and connect it to c_1 . Connect h_i^3 to $h_i^1, h_i^{1'}, h_i^2,$ and $h_i^{2'}$. Finally, connect h_i to $h_i^1, h_i^{1'}, h_i^2, h_i^{2'},$ and h_i^3 . Let H be the set of all vertices created in this step. These vertices will ensure that for each i , there is exactly one vertex $p_{i,j}$ of color 1.
 6. Add a vertex v_j for all $j \in [n]$ and connect v_j to $p_{i,j}$ for all $i \in [k]$. Add vertices v_j^1 and $v_j^{1'}$ and connect them to c_1 and c_3 . Add vertices v_j^2 and $v_j^{2'}$ and connect these to c_1 and c_2 . Add a vertex v_j^3 . Finally, connect vertex v_j to $v_j^1, v_j^{1'}, v_j^2, v_j^{2'},$ and v_j^3 . The vertices added in this step ensure that there cannot be $i, i' \in [k]$ such that both $p_{i,j}$ and $p_{i',j}$ receive color 1 for some $j \in [n]$.
 7. For each $i, i' \in [k]$ and $j, j' \in [n]$, add vertex $s_{j,j'}^{i,i'}$; let the set containing all these vertices be S . Connect $s_{j,j'}^{i,i'}$ to $p_{i,j}$ and $p_{i',j'}$. Furthermore, connect $s_{j,j'}^{i,i'}$ to y_ℓ whenever $\{j, j'\}$ is *not* an edge in instance X_ℓ . These vertices are used to verify whether the vertices selected by P indeed form a clique in the selected input instance.
 8. For each $i, i' \in [k]$ and $j, j' \in [n]$, add vertices $f_{j,j'}^{i,i'}, g_{j,j'}^{i,i'}, b_{j,j'}^{i,i'},$ and $b_{j,j'}^{i,i'}$, and connect all these vertices to $s_{j,j'}^{i,i'}$. Connect $g_{j,j'}^{i,i'}$ to c_1 and c_3 , connect $b_{j,j'}^{i,i'}$ to c_1 and c_2 , and finally connect $b_{j,j'}^{i,i'}$ to c_1 and c_2 .
 9. For every vertex in $v \in V(G) \setminus V(C_q)$, add the edges $\{v, c_i\}$ and $\{v, c'_i\}$ for all $3 < i \leq q$. Thus, we connect every nonpalette vertex in G to all but the first three colors from the palette. This step ensures that colors $i > 3$ are not used to color $V(G) \setminus C_q$.

It follows from Lemma 4.14 that all c_i receive distinct colors. Therefore, we will from now on assume that $c(c_i) = i$ for any coloring c . Furthermore, we observe that for $i \in [q]$, vertex c_i is connected to c'_j and c_j for all $j \in [q] \setminus \{i\}$. It follows that vertex c_i has its own color (if any) as its CNCF-color, since it is connected to two vertices of all remaining colors.

In the proofs of the remaining claims, we will regularly use that any nonpalette vertex in G has two neighbors of color i for all $i > 3$.

CLAIM 4.17. *For any q -CNCF-coloring c of G , there exists exactly one vertex $y^* \in Y$ such that $c(y^*) = 2$.*

Proof. It follows from the observation above, that $c(a_1) = c(a'_1) = 1$ and $c(a_2) = c(a'_2) = 2$. Furthermore, $c(a_3) \neq 2$. Thereby, $N[a_3]$ contains vertex a , together with one vertex of color 2 and two vertices of color i for all $i \neq 2$, implying $c(a) \neq 2$. It follows that $N[a]$ contains at least two vertices of color 1 and two of color 3 and that $N[a] \setminus Y$ contains no vertices of color 2. Thereby, $N[a] \cap Y = Y$ must have exactly one vertex of color 2. J

CLAIM 4.18. *For any 3-CNCF-coloring c of G , there exist distinct j_1, \dots, j_k such that $c(p_{i,j_i}) = 1$ and for all other $p \in P$, $c(p) \neq 1$.*

Proof. We start by showing that for each $i \in [k]$, there exists $j_i \in [n]$ such that $c(p_{i,j_i}) = 1$. We will then show that these j_i are indeed distinct.

Let $i \in [k]$, then $\{h_i^1, h_i'^1, h_i^2, h_i'^2\} \subseteq N[h_i]$ and thus $N[h_i]$ contains two vertices of colors 2 and 3. Since h_i is connected to h_i^3 and $N[h_i^3]$ contains at least one vertex of color 1, and two vertices of both color 2 and 3, it follows that $c(h_i) \neq 1$. Thereby, the CNCF-color for h_i is 1 and thus there exists a unique vertex in $\{p_{i,j} \mid j \in [n]\}$ that receives color 1.

It remains to show that these j_i are indeed distinct. We do this by showing that there cannot be vertices $p_{i,j}$ and $p_{i',j}$ such that $c(p_{i,j}) = c(p_{i',j}) = 1$. Suppose for contradiction that there are $j \in [n]$ and $i, i' \in [k]$ such that $c(p_{i,j}) = c(p_{i',j}) = 1$. But then $N[v_j]$ contains vertices v_j^1 and $v_j'^1$ that have color 2, vertices v_j^2 and $v_j'^2$ of color 3, and the aforementioned two vertices of color 1. Since it furthermore contains two vertices of color i for all $i \geq 4$, this contradicts that c is a CNCF-coloring for G . \square

Using the claims above, we now prove the correctness of this cross-composition.

CLAIM 4.19. *If there exists $\ell \in [t]$ such that X_ℓ has a clique of size k , then G is q -CNCF-colorable.*

Proof. Take ℓ such that X_ℓ is a yes-instance for CLIQUE and let j_1, \dots, j_k be such that vertices $\{j_1, \dots, j_k\}$ form this clique in instance X_ℓ . We give a coloring $c : V(G) \rightarrow [q]$ of G . We start by showing how to color the vertices defined in each step of the construction; this coloring is also depicted in Figure 5.

1. We start by coloring the palette C in Lemma 4.15, such that $c(c_i) := c(c'_i) := i$.
2. Let $c(y_\ell) := 2$ and $c(y) := 3$ for all other vertices $y \in Y$.
3. Let $c(a) := c(a_1) := c(a'_1) := 1$ and $c(a_2) := c(a'_2) := c(a_3) := 3$.
4. For all $i \in [k]$, let $c(p_{i,j_i}) := 1$. For all other $p \in P$, let $c(p) := 3$.
5. For all $i \in [k]$, let $c(h_i) := c(h_i^1) := c(h_i'^1) := c(h_i^3) := 2$ and let $c(h_i^2) := c(h_i'^2) := 3$.
6. For all $j \in [n]$, let $c(v_j) := 3$. For all $j \in [n]$, let $c(v_j^1) := c(v_j'^1) := 2$ and let $c(v_j^2) := c(v_j'^2) := 3$. Let $c(v_j^3) := 2$ if there exists $i \in [k]$ such that $j_i = j$. Let $c(v_j^3) := 1$ otherwise.
7. For $i, i' \in [k]$ and $j, j' \in [n]$, let $c(s_{j,j'}^{i,i'}) := 3$.
8. For $i, i' \in [k]$ and $j, j' \in [n]$, let $c(g_{j,j'}^{i,i'}) := 2$, $c(b_{j,j'}^{i,i'}) := c(b_{j,j'}'^{i,i'}) := 3$. Finally, if $s_{j,j'}^{i,i'}$ at this point has no neighbor of color 1, let $c(f_{j,j'}^{i,i'}) := 1$. Furthermore, if $s_{j,j'}^{i,i'}$ is not connected to y_ℓ (meaning $\{j, j'\}$ is an edge in X_ℓ), define $c(f_{j,j'}^{i,i'}) := 1$.
1. Otherwise, let $c(f_{j,j'}^{i,i'}) := 2$.

It remains to show that this indeed gives a q -CNCF-coloring of G . We verify this for all vertices. C is CNCF-colored by the fact that c_i and c'_i are colored by their own color and not connected to any other vertex of color i . Vertices in Y are CNCF-colored by vertex a which has color 1. $N[a]$ contains exactly one vertex of color 2, namely, y_ℓ . $N[a_1], N[a'_1], N[a_2], N[a'_2]$, and $N[a_3]$ all contain exactly one vertex of color 2, namely, c_2 . For $p \in P$, $N[p]$ contains exactly one vertex h_i of color 2 and no other vertices of color 2. For all $i \in [k]$, $N[h_i]$ contains exactly one vertex p_{i,j_i} of color 1 and no other vertices of color 1. Vertices $h_i^1, h_i'^1, h_i^2, h_i'^2$, and h_i^3 have c_1 as their unique neighbor with color 1. Similarly, for all $j \in [n]$, the vertex v_j has exactly one neighbor

of color 1 from the set $\{p_{i,j} \mid i \in [k]\} \cup \{v_i^3\}$. Vertices $v_j^1, v_j^{1'}, v_j^2$, and $v_j^{2'}$ have c_1 as their only neighbor of color 1. Vertex v_j^3 has a distinct color from its only neighbor v_j and thereby its closed neighborhood is CNCF-colored. Since $c(f_{j,j'}^{i,i'}) \in \{1, 2\}$ and $c(s_{j,j'}^{i,i'}) = 3$ for all $i, i' \in [k], j, j' \in [n]$, vertex $f_{j,j'}^{i,i'}$ receives a different color than its only neighbor. Vertices $g_{j,j'}^{i,i'}, b_{j,j'}^{i,i'}$, and $b_{j,j'}^{i',i'}$ all have a unique neighbor of color 1, namely, vertex c_1 . Finally we check the closed neighborhood of vertices $s_{j,j'}^{i,i'}$ for $i, i' \in [k], j, j' \in [n]$. If $s_{j,j'}^{i,i'}$ is not connected to y_ℓ , it is ensured that it has exactly one neighbor of color 2, namely, vertex $g_{j,j'}^{i,i'}$. Otherwise, observe that $c(p_{i,j}) \neq 1$ or $c(p_{i',j'}) \neq 1$ as $\{j, j'\}$ is not an edge in X_ℓ . The choice of coloring for $f_{j,j'}^{i,i'}$ ensures that in this case, $s_{j,j'}^{i,i'}$ has a unique neighbor of color 1. \lrcorner

CLAIM 4.20. *If G has a q -CNCF-coloring, then there exists $\ell \in [t]$ such that X_ℓ has a clique of size k .*

Proof. Let c be a CNCF-coloring of G . It follows from Claim 4.17 that there exists a vertex $y \in Y$ with $c(y) = 2$. Let ℓ be such that $c(y_\ell) = 2$. We show that X_ℓ has a clique of size k . By Claim 4.18, there exist distinct j_1, \dots, j_k such that $c(p_{i,j_i}) = 1$. We show that the vertices j_1, \dots, j_k form the desired clique in X_ℓ .

Suppose for contradiction that there are distinct $i, i' \in [k]$ such that $\{j_i, j_{i'}\}$ is not an edge in instance X_ℓ . We will show that $N[s_{j_i, j_{i'}}^{i, i'}]$ is not properly CNCF-colored. First of all, $N[s_{j_i, j_{i'}}^{i, i'}]$ contains the two vertices p_{i, j_i} and $p_{i', j_{i'}}$ with $c(p_{i, j_i}) = c(p_{i', j_{i'}}) = 1$. Furthermore it contains the two vertices $b_{j_i, i'}^{i, i'}$ and $b_{j_{i'}, i'}^{i', i}$ that have color 3, and finally it contains two vertices of color 2, namely, $g_{j_i, i'}^{i, i'}$ and y_ℓ . Furthermore, $N[s_{j_i, j_{i'}}^{i, i'}]$ contains two vertices of color i for all $i > 3$, by step 9 of the construction. This however contradicts that c is a CNCF-coloring of G , and thus we conclude that j_1, \dots, j_k form a clique of size k in instance X_ℓ . \lrcorner

It follows from Claims 4.19 and 4.20 that G has a q -CNCF-coloring if and only if one of the given input instances was a yes-instance for CLIQUE. It remains to bound the size of a vertex cover in G , to conclude the cross-composition. It is easy to verify that $V(G) \setminus Y$ is a vertex cover for G , since Y is an independent set. Thereby the size of a vertex cover in G is at most $|V(G) \setminus Y| = \mathcal{O}(n^2k^2) + f(q)$, where $f(q)$ is the size of palette-gadget C_q . As this is properly bounded for a cross-composition, the theorem statement follows from Theorem 2.10. \square

4.3. Generalized kernel for 2-CNCF-COLORING. In this part we prove Theorem 4.2, by obtaining a polynomial generalized kernel for 2-CNCF-COLORING parameterized by vertex cover size. This result is in contrast to the kernelization results we obtain for q -CNCF-COLORING for $q \geq 3$ as well as q -ONCF-COLORING for $q \geq 2$. We will start by transforming an instance of 2-CNCF-COLORING to an equivalent instance of another problem, namely, d -POLYNOMIAL ROOT CSP. We will then carefully rephrase the d -POLYNOMIAL ROOT CSP instance such that it uses only a limited number of variables, such that we can use a known kernelization result for d -POLYNOMIAL ROOT CSP to obtain our desired compression. We start by introducing the relevant definitions.

Define d -POLYNOMIAL ROOT CSP over a field F as follows [16].

***d*-POLYNOMIAL ROOT CSP**

Input: A list L of polynomial equalities over variables $V = \{x_1, \dots, x_n\}$. An equality is of the form $f(x_1, \dots, x_n) = 0$, where f is a multivariate polynomial over F of degree at most d .

Question: Does there exist an assignment of the variables $\tau: V \rightarrow \{0, 1\}$ satisfying all equalities (over F) in L ?

A field F is said to be *efficient* if both the field operations and Gaussian elimination can be done in polynomial time in the size of a reasonable input encoding. In particular, \mathbb{Q} is an efficient field by this definition. The following theorem was shown by Jansen and Pieterse.

THEOREM 4.21 (see [16, Theorem 3.1]). *There is a polynomial-time algorithm that, given an instance (L, V) of *d*-POLYNOMIAL ROOT CSP over an efficient field F , outputs an equivalent instance (L', V) with at most $n^d + 1$ constraints such that $L' \subseteq L$.*

Using the theorem introduced above, we can now prove Theorem 4.2.

Proof of Theorem 4.2. Given an input instance G with vertex cover S of size k , we start by preprocessing G . For each set $X \subseteq S$ with $|X| \leq 2$, mark 3 vertices in $v \in G \setminus S$ with $N(v) = X$ (if there do not exist 3 such vertices, simply mark all). Let $S' \subseteq V(G) \setminus S$ be the set of all marked vertices. Remove all $w \in V(G) \setminus (S \cup S')$ with $\deg(w) \leq 2$ from G . Let the resulting graph be G' .

CLAIM 4.22. *G' is 2-CNCF-colorable if and only if G is 2-CNCF-colorable.*

Proof. In one direction, suppose G' has a 2-CNCF coloring c using colors $\{r, b\}$. Consider a vertex $w \in V(G) \setminus V(G')$. Let $X_w \subseteq S$ be the neighborhood of w . Note that $|X_w|$ is at most 2. Consider $N(X_w) \cap S'$. Since w was deleted, there are 3 vertices in $N(X_w) \cap S'$. Consider the color from $\{r, b\}$ that appears in the majority on the vertices of $N(X_w) \cap S'$. If we color w with the same color, it is easy to verify that this extension of c to G is a 2-CNCF coloring of G .

In the reverse direction, suppose G has a 2-CNCF coloring c using colors $\{r, b\}$. We describe a new coloring c' for G as follows. Consider a subset $X \subseteq S$ of size at most 2 and let N be the set of vertices in $G \setminus S$ that have X as their neighborhood. If $|N| > 3$ and $N \setminus S'$ has a vertex w that is uniquely colored in the set N , then we arbitrarily choose a vertex $w' \in N \cap S'$. We define $c'(w') = c(w)$ and $c'(w) = c(w')$. All other vertices have the same color in c and c' . It is easy to verify that c' is also a 2-CNCF coloring of G and the restriction of c' to G' is a 2-CNCF coloring of G' . \square

We continue by creating an instance of 2-POLYNOMIAL ROOT CSP that is satisfiable if and only if G' is 2-CNCF-colorable. Let $V := \{r_v, b_v \mid v \in V(G)\}$ be the variable set. We create L over \mathbb{Q} as follows.

1. For each $v \in V(G')$, add the constraint $r_v + b_v - 1 = 0$ to L .
2. For all $v \in V(G')$, add the constraint

$$\left(-1 + \sum_{u \in N[v]} r_u\right) \cdot \left(-1 + \sum_{u \in N[v]} b_u\right) = 0.$$

3. For each $v \in V(G') \setminus (S \cup S')$ of degree $d_v = |N(v)|$ add the constraint

$$\left(\sum_{u \in N(v)} r_u\right) \left(-1 + \sum_{u \in N(v)} r_u\right) \left(-d_v - 1\right) + \sum_{u \in N(v)} r_u \left(-d_v + \sum_{u \in N(v)} r_u\right) = 0.$$

Note that such a constraint is a polynomial of degree at most 4.

Intuitively, the first constraint ensures that every vertex is either red or blue. The second constraint ensures that in the closed neighborhood of every vertex, exactly one vertex is red or exactly one is blue. The third constraint is (currently) redundant, saying that the open neighborhood of every vertex outside the vertex cover does not have two red or two blue vertices, which is clearly forbidden. We will need this requirement later, in the proof of Claim 4.26.

We show that this results in an instance that is equivalent to the original input instance, in the following sense.

CLAIM 4.23. *(L, V) is a yes-instance of 2-POLYNOMIAL ROOT CSP if and only if G' is 2-CNCF-colorable.*

Proof. Suppose $\tau : V \rightarrow \{0, 1\}$ is a satisfying assignment for (L, V) . We show how to define a 2-CNCF coloring $c : V \rightarrow \{red, blue\}$ for G' . Let $c(v) := red$ if $\tau(r_v) = 1$ and let $c(v) := blue$ if $\tau(b_v) = 1$. Note that this defines exactly one color for each vertex, as by step 1, $r_v + b_v = 1$ and we used at most two distinct colors. It remains to show that this is indeed a CNCF-coloring. Let $v \in V(G')$ be an arbitrary vertex; we show that $N[v]$ is conflict-free colored. It follows from the equations added in step 2, that one of the following holds.

- $(\sum_{u \in N[v]} \tau(r_u) = 1)$. In this case, $N[v]$ contains exactly one vertex $u \in N[v]$ with $c(u) = red$, showing that $N[v]$ is conflict-free colored.
- $(\sum_{u \in N[v]} \tau(b_u) = 1)$. In this case, $N[v]$ contains exactly one vertex $u \in N[v]$ with $c(u) = blue$, showing that $N[v]$ is conflict-free colored.

This concludes this direction of the proof.

For the other direction, suppose G' has 2-CNCF-coloring c , we show how to define a satisfying assignment τ for (L, V) . For $v \in V(G')$, let $\tau(r_v) := 1$ if $c(v) = red$ and let $\tau(r_v) := 0$ otherwise. Similarly, $\tau(b_v) := 1$ if $c(v) = blue$ and let $\tau(b_v) := 0$ otherwise. Observe that by this definition, $\tau(r_v) = 1 - \tau(b_v)$ for all $v \in V(G)$, showing that we satisfy all equations introduced in step 1. For the equations introduced in step 2, consider an arbitrary vertex $v \in V(G')$. Suppose its CNCF-color is red, then $N[v]$ contains exactly one vertex u with $c(u) = red$ and thus $\sum_{u \in N[v]} r_u = 1$, implying $(-1 + \sum_{u \in N[v]} r_u) \cdot (-1 + \sum_{u \in N[v]} b_u - 1) = 0$ as desired. Similarly, if its CNCF-color is blue we obtain $\sum_{u \in N[v]} b_u = 1$ and again $(-1 + \sum_{u \in N[v]} r_u) \cdot (-1 + \sum_{u \in N[v]} b_u) = 0$. It remains to prove that the equations added in step 3 are satisfied. For this, let v be an arbitrary vertex for which the equation was added. Observe that if v is colored red, then its neighborhood contains no red vertices, such that $\sum_{u \in N(v)} r_u = 0$ and the equation is satisfied, or $d - 1$ red vertices, such that $\sum_{u \in N(v)} r_u = d - 1$ and again the equation is satisfied. If v is colored blue, then either its neighborhood is entirely red, such that $\sum_{u \in N(v)} r_u = d$ or it contains exactly one red vertex, such that $\sum_{u \in N(v)} r_u = 1$. In both cases the equation is satisfied. \lrcorner

Clearly, $|V| = 2n$ if n is the number of vertices of G' . We will now show how to modify L , such that it uses only variables for the vertices in $S \cup S'$. To this end, we introduce the following function. For $v \notin (S \cup S')$, let $f_v(V) := g(\sum_{u \in N(v)} r_u, |N(v)|)$, where

$$g(x, N) = -\frac{(N - x)(x - 1)(N - 2(x + 1))}{N(N - 2)}.$$

Note that for any fixed $N > 2$, $g(x, N)$ describes a degree-3 polynomial in x over \mathbb{Q} . The following is easy to verify.

OBSERVATION 4.24. $g(0, N) = g(N - 1, N) = 1$, and $g(N, N) = g(1, N) = 0$ for all $N \in \mathbb{Z} \setminus \{0, 2\}$.

Observe that f_v only uses variables defined for vertices that are in S . As such, let $V' := \{r_v, b_v \mid v \in S\} \cup \{r_v, b_v \mid v \in S'\}$, and let L' be equal to L with every occurrence of r_v for $v \notin (S \cup S')$ substituted by f_v and every occurrence of b_v for $v \notin (S \cup S')$ substituted by $(1 - f_v(V))$.

CLAIM 4.25. *If $\tau: V \rightarrow \{0, 1\}$ is a satisfying assignment for (L, V) , then $\tau|_{V'}$ is a satisfying assignment for (L', V') .*

Proof. We show this by showing that for all $v \notin (S \cup S')$, $f_v(\tau(V)) = \tau(r_v)$ in this case. Since $\tau(b_v) = 1 - \tau(r_v)$ by the constraints added in step 1, this will conclude the proof. Consider an arbitrary vertex $v \notin S$. Observe that by the equations added in step 2, we are in one of the following cases.

- $\sum_{u \in N[v]} \tau(r_u) = 1$ and $\tau(r_v) = 1$. In this case, $\sum_{u \in N(v)} \tau(r_u) = 0$ and thereby $f_v(V) = g(0, |N(v)|) = 1 = \tau(r_v)$ by Observation 4.24.
- $\sum_{u \in N[v]} \tau(r_u) = 1$ and $\tau(r_v) = 0$. In this case, $\sum_{u \in N(v)} \tau(r_u) = 1$ and thereby $f_v(V) = g(1, |N(v)|) = 0 = \tau(r_v)$ using Observation 4.24.
- $\sum_{u \in N[v]} \tau(b_u) = 1$ and $\tau(b_v) = 1$. In this case, since $\tau(b_v) = 1 - \tau(r_v)$ for all v , we obtain that $\sum_{u \in N(v)} \tau(b_u) = 0$, and thus $\sum_{u \in N(v)} \tau(r_u) = |N(v)|$. Thereby, $f_v(V) = g(|N(v)|, |N(v)|) = 0 = 1 - \tau(b_v) = \tau(r_v)$ by Observation 4.24.
- $\sum_{u \in N[v]} \tau(b_u) = 1$ and $\tau(b_v) = 0$. Hereby, $\sum_{u \in N(v)} \tau(b_u) = 1$ and thus $\sum_{u \in N[v]} \tau(r_u) = |N(v)| - 1$. Thus, $f_v(V) = g(|N(v)| - 1, |N(v)|) = 1 = 1 - \tau(b_v) = \tau(r_v)$ using Observation 4.24. \lrcorner

The next claim shows the equivalence of (L', V') and (L, V) .

CLAIM 4.26. *If $\tau: V' \rightarrow \{0, 1\}$ is a satisfying assignment for (L', V') , then there exists a satisfying assignment $\tau': V \rightarrow \{0, 1\}$ for (L, V) such that $\tau'|_{V'} = \tau$.*

Proof. Let τ be given, we show how to construct τ' . For all $x \in V'$, let $\tau'(x) := \tau(x)$. Furthermore, for $r_v \in V \setminus V'$, let $\tau(r_v) := f_v(\tau(V))$ and let $\tau(b_v) := 1 - \tau(r_v)$. Since L' was simply obtained from L by substituting r_v by $f_v(V)$ and b_v by $(1 - f_v(V))$ in all constraints, it is clear that τ' satisfied all equations in L' . It remains to show that $\tau(r_v) \in \{0, 1\}$ for all $v \in V$. If $v \in V'$, this is obvious, so suppose $v \in V \setminus V'$ such that $\tau(r_v) = f_v(\tau(V))$. Observe that an equation was added for v in step 3. Therefore, we know that $\sum_{u \in N(v)} r_u \in \{0, 1, d_v - 1, d_v\}$ and it follows from Lemma 4.24 that $f_v(\tau(V))$ takes a Boolean value. \lrcorner

Using the method described above, we obtain an instance (L, V) of 2-POLYNOMIAL ROOT CSP such that (L, V) has a satisfying assignment if and only if G is 2-CNCF-colorable by Claims 4.22 and 4.23. Then we obtain an instance (L', V') such that (L', V') is satisfiable if and only if (L, V) is satisfiable by Claims 4.25 and 4.26. As such, (L', V') is a yes-instance if and only if G is 2-CNCF-colorable and it suffices to give a kernel for (L', V') . Observe that $|V'| = \mathcal{O}(k^2)$.

We start by partitioning L' into three sets L'_S , L'_1 , and L'_2 . Let L'_S contain all equalities created for a vertex $v \in S$. Let L'_1 contain all equations that contain at least one of the variables in $\{r_v, b_v \mid v \in S'\}$ and let L'_2 contain the remaining equalities. Observe that $|L'_S| = k$ by definition. Furthermore, the polynomials in L'_1 have degree at most 2, as they were created for vertices in $V(G') \setminus S$, and these are not connected. As such, we use Theorem 4.21 to obtain $L''_1 \subseteq L'_1$ such that $|L''_1| = \mathcal{O}((k^2)^2) = \mathcal{O}(k^4)$ and any Boolean assignment satisfying all equalities in L''_1 satisfies all equalities in L'_1 .

Similarly, we observe that L'_2 by definition contains none of the variables in $\{r_v, b_v \mid v \in S'\}$, implying that the equations in L'_2 are equations over only k variables. Since the polynomials in L'_2 have degree at most 6, we can apply Theorem 4.21 to obtain $L''_2 \subseteq L'_2$ such that $|L''_2| \leq \mathcal{O}(k^6)$ and any assignment satisfying all equations in L''_2 satisfies all equalities in L'_2 .

We now define $L'' := L''_1 \cup L''_2 \cup L'_S$, and the output of our polynomial generalized kernel will be (L'', V') . The correctness of the procedure is proven above, it remains to bound the number of bits needed to store instance (L'', V') .

By this definition, $|L''| \leq \mathcal{O}(k^6)$. To represent a single constraint, it is sufficient to store the coefficients for each variable in V' . The storage space needed for a single coefficient is $\mathcal{O}(\log(n))$, as the coefficients are bounded by a polynomial in n . Thereby, (L'', V') can be stored in $\mathcal{O}(k^6 \cdot k^2 \log n)$ bits. To bound this in terms of k , we observe that we can solve 2-CNCF-COLORING in time $\mathcal{O}(8^k \cdot \text{poly}(n))$ by Theorem 3.1, using that the treewidth of a graph is bounded by its vertex cover size. Therefore, we can assume that $\log(n) \leq k$, as otherwise we can solve the 2-CNCF-COLORING problem in $\mathcal{O}(8^k \cdot \text{poly}(n))$ time, which is then polynomial in n . Thereby we conclude that (L'', V') can be stored in $\mathcal{O}(k^9)$ bits. \square

4.4. Kernelization bounds for conflict-free coloring extension. We continue by studying a specific type of precoloring extension, where a partial conflict-free coloring is given and needs to be extended to the entire graph. Generally speaking, such a problem would be more general (and thus harder) than the original problem, but in this case we will require that a vertex cover of the graph is precolored such that it remains to color an independent set. More precisely, the studied extension problems are defined as follows.

q -CNCF-COLORING-VC-EXTENSION

Input: A graph G with vertex cover S and partial q -coloring $c: S \rightarrow [q]$.

Question: Does there exist a q -CNCF-coloring of G that extends c ?

We define q -ONCF-COLORING-VC-EXTENSION analogously.

We obtain the following kernelization results when parameterized by vertex cover size, thereby classifying the situations where the extension problem has a polynomial kernel. The extension problem turns out to have a polynomial kernel in the same case as the normal problem. However, we manage to give a significantly smaller kernel. Observe that the kernelization result is nontrivial, since 2-CNCF-COLORING-VC-EXTENSION is NP-hard (see Theorem 4.30 below).

THEOREM 4.27. *The following results hold.*

1. 2-CNCF-COLORING-VC-EXTENSION has a kernel with $\mathcal{O}(k^2)$ vertices and edges that can be stored in $\mathcal{O}(k^2 \log k)$ bits. Here k is the size of the input vertex cover S .
2. q -CNCF-COLORING-VC-EXTENSION for any $q \geq 3$, and q -ONCF-COLORING-VC-EXTENSION for any $q \geq 2$ parameterized by the size of a vertex cover, do not have a polynomial kernel, unless $\text{NP} \subseteq \text{coNP}/\text{poly}$.

We start by noting that the kernelization lower bounds given in the previous sections still apply. In particular, we obtain the following two corollaries.

COROLLARY 4.28. *For any $q \geq 2$, q -ONCF-COLORING-VC-EXTENSION parameterized by the size of a vertex cover does not have a polynomial kernel, unless $\text{NP} \subseteq \text{coNP}/\text{poly}$.*

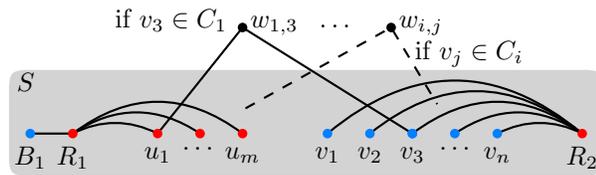


FIG. 6. The reduction from MONOTONE EXACT SAT to 2-CNCF-COLORING-VC-EXTENSION.

Proof. Observe that in the cross-composition given in Lemma 4.3, the vertices

$$H \cup \{R, R', B, B', a, a_1, a_2, a'_1, a'_2\} \cup \{v_j, v_j^1, v_j^{i1}, v_j^{i2}, v_j^{i'2} \mid j \in [n]\} \\ \cup S \cup \{\text{all vertices in gadgets labeled } g_{10}, g_8, g_4, \text{ or } g_5\}$$

form a vertex cover of appropriately bounded size, and that these vertices always receive the same color in the proof of Claim 4.8. Furthermore, in the linear parameter transformation given in Lemma 4.10, we see that all additional added vertices always receive the same color in the proof of Claim 4.12. Hence, all these vertices could be safely precolored and the bounds still hold. \square

Furthermore, Lemma 4.16 immediately gives us the following result on this extension problem.

COROLLARY 4.29. *For any $q \geq 3$, q -CNCF-COLORING-VC-EXTENSION parameterized by the size of a vertex cover does not have a polynomial kernel, unless $\text{NP} \subseteq \text{coNP/poly}$.*

Proof. The result follows immediately from the same cross-composition as given in the proof of Lemma 4.16. Observe that the vertices $C \cup H \cup \{v_j \mid j \in [n]\} \cup \{s_{j,j'}^{i,i'} \mid i, i' \in [k], j, j' \in [n]\} \cup \{a, a_1, a'_1, a_2, a'_2, a_3\}$ form a vertex cover of the created graph G of size $\text{poly}(n)$, and that they are always given the same coloring in the proof of Claim 4.19. \square

The results above prove part 2 of Theorem 4.27. We will now show that 2-CNCF-COLORING-VC-EXTENSION has a simple polynomial kernel of size $\mathcal{O}(k^2 \log k)$, where k is the size of the vertex cover. This proves part 1 of Theorem 4.27. We start by arguing that 2-CNCF-COLORING-VC-EXTENSION is indeed NP-hard.

THEOREM 4.30. *2-CNCF-COLORING-VC-EXTENSION is NP-hard.*

Proof. We prove this by a reduction from MONOTONE EXACT SAT, which is defined as follows.

MONOTONE EXACT SAT

Input: A formula \mathcal{F} over variable set X that is a conjunction of clauses, where each clause consists of a number of variables from X .

Question: Does there exist an assignment $\tau: X \rightarrow \{0, 1\}$ such that every clause in \mathcal{F} contains exactly one variable that is set to 1?

It is known that the MONOTONE EXACT SAT problem is NP-hard, as it generalizes problem NP1 in [19]. Let an instance $\mathcal{F} = C_1 \wedge \dots \wedge C_m$ over variables $X = \{x_1, \dots, x_n\}$ be given, we show how to construct a graph G with vertex cover S and precoloring $f: S \rightarrow \{\text{red}, \text{blue}\}$ for 2-CNCF-COLORING-VC-EXTENSION. See Figure 6 for a sketch of G .

- Add vertices R_1, R_2 , and B_1 to G and to S . Let $c(R_1) := c(R_2) := \text{red}$ and let $c(B_1) := \text{blue}$. Connect R_1 to B_1 .

- For every clause C_i in \mathcal{F} , add a vertex u_i to G . Let u_i be contained in S and let $c(u_i) := red$. Connect all vertices u_i to R_2 .
- For every variable $x_j \in X$, add a vertex v_j to G , let v_j be contained in S , and set $c(v_j) := blue$. Connect v_j to R_2 .
- For every $i \in [m], j \in [n]$, if variable x_j is in clause C_i , construct a new vertex $w_{i,j}$ and connect $w_{i,j}$ to v_j and u_i . $w_{i,j}$ is not contained in S .

Clearly, S is hereby a vertex cover of G and it is colored by c . It is easy to see that the construction above can be done in polynomial time. It remains to show that c can be extended to a 2-CNCF-coloring of G if and only if \mathcal{F} is satisfiable.

(\Rightarrow) Suppose \mathcal{F} is satisfiable and has satisfying assignment $\tau: X \rightarrow \{0, 1\}$, we give a 2-CNCF-coloring $c': V(G) \rightarrow \{red, blue\}$ of G , such that c' extends c . Naturally, for every vertex $s \in S$, let $c'(s) := c(s)$. For $i \in [m], j \in [n]$, let $c(w_{i,j}) := blue$ if $\tau(x_j) = 1$ and let $c(w_{i,j}) := red$ otherwise. It remains to show that this is a valid CNCF-coloring of G . We check the neighborhoods of all vertices.

$N[R_1] = \{R_1, B_1\} \cup \{u_i \mid i \in [m]\}$. Here B_1 is a unique blue vertex. $N[R_2] = \{R_2\} \cup \{v_j \mid j \in [n]\}$ and R_2 is a unique red vertex in this set. $N[B_1] := \{B_1, R_1\}$ and these vertices are red and blue, as desired. For any vertex u_i , $N[u_i]$ contains one blue vertex, namely, $w_{i,j}$ where j is such that x_j is the *unique* variable in clause C_i with $\tau(x_j) = 1$. For any $j \in [n]$, $N[v_j] := \{w_{i,j} \mid x_j \in C_i\} \cup \{v_j, R_2\}$. Since all vertices in $\{w_{i,j} \mid x_j \in C_i\}$ receive the same color, this set has a uniquely colored vertex which is either v_j (which is blue) or R_2 (which is red). For any $i \in [m], j \in [n]$, vertex $w_{i,j}$ has exactly two neighbors and these receive different colors, and thus $N[w_{i,j}]$ has a neighbor with a unique color.

(\Leftarrow) Let $c': V(G) \rightarrow \{red, blue\}$ be a CNCF-Coloring of G . Let $j \in [n]$, then $N[v_j] = \{w_{i,j} \mid v_j \in C_i\} \cup \{R_2, v_j\}$. Thereby, we observe that the vertices in $W_j := \{w_{i,j} \mid v_j \in C_i\}$ all receive the same color since $c'(v_j) = blue$ and $c'(R_2) = red$. Let $\tau(x_j) := 1$ if all vertices in W_j have color blue, and let $\tau(x_j) := 0$ otherwise. We show that τ is a satisfying assignment for \mathcal{F} . Let C_i be a clause of \mathcal{F} , we show that there is exactly one x_j such that $\tau(x_j) = 1$, by showing that there is exactly one $j \in [n]$ such that vertex $c'(w_{i,j}) = blue$. Consider vertex u_i , then $N[u_i] = \{u_i, R_1\} \cup \{w_{i,j} \mid v_j \in C_i\}$. Since $c'(u_i) = c'(R_1) = red$ since c' extends c , it trivially follows that there is indeed a unique j such that $c'(w_{i,j}) = blue$. This concludes the proof. \square

We now show that, unlike 3-CNCF-COLORING-VC-EXTENSION, 2-CNCF-COLORING-VC-EXTENSION has a simple polynomial kernel.

LEMMA 4.31. 2-CNCF-COLORING-VC-EXTENSION *parameterized by the size of the vertex cover has a kernel of size $\mathcal{O}(k^2 \log k)$.*

Proof. Let G with partial coloring c and vertex cover S be an instance of the problem with $|S| \leq k$. We first show how to obtain an equivalent instance (G', S', c') such that $|S'| \leq 3|S|$ and such that every vertex in $V(G') \setminus S'$ has degree at most two. Then we can use a procedure given by Gargano and Rescigno [11] to further reduce the number of vertices in $V(G') \setminus S'$ to at most $\mathcal{O}(|S'|^2) = \mathcal{O}(k^2)$.

If there exists a vertex that has at least two red and two blue neighbors by this precoloring, output a trivial no-instance. For the rest of the kernelization, we can thus assume that this case does not occur. Initialize (G', S', c') as (G, S, c) . Observe that for any vertex of degree at least 3, its coloring is now completely determined. While there exists a vertex v of degree at least three in $G' \setminus S'$, we define $c'(v)$ as follows. If v has only red neighbors, let $c'(v) := blue$. Otherwise, if v has at least two red and exactly one blue neighbor, let $c'(v) := red$. Similarly, if v has only blue neighbors let $c'(v) := red$ and if v has exactly one red neighbor let $c'(v) := blue$. It is

easy to see that there is a 2-CNCF coloring of G that extends c , if and only if there is one extending c' .

For each vertex $u \in S'$, mark two neighbors that are colored red by c' , and two that are blue (if these exist). Add all marked vertices to S' , and delete all vertices $v \in G \setminus S'$ that have degree at least three and are not marked. Observe that hereby c' is a coloring of the vertices of S' , S' is a vertex cover of G' , and each vertex in $V(G') \setminus S'$ has degree at most two. We argue the following.

CLAIM 4.32. *The graph G' has a 2-CNCF-coloring extension if and only if G has a 2-CNCF-coloring extension.*

Proof. Suppose G has a 2-CNCF-coloring extension of c ; by the observation above there is also a 2-CNCF-extension of c' , let this be c'' . We show that $c''|_{V(G')}$ is a proper coloring of G' . Every vertex in $G' - S$ has the same neighborhood as in G , and thus this neighborhood is conflict-free colored by c'' . For every vertex in $s \in S$, $N[s] \cap (S \cup \{v \notin S \mid d(v) \leq 2\})$ is the same in G and G' . For the vertices in $S \cup \{v \notin S \mid d(v) = 2\}$ the color is the same for any 2-CNCF-coloring of G and we kept two red and two blue vertices. As such, c'' is a CNCF-coloring of G' .

Suppose G' has a 2-CNCF-coloring extension c'' of c' . We define a 2-CNCF-coloring d of G that extends c . Start by defining $d(v) = c''(v)$ for any vertex $v \in V(G) \cap V(G')$. Hereby, all vertices in the vertex cover S of G are colored. Let $v \in V(G) \setminus S$. Note that v has at least three neighbors in s , as otherwise v would have been a vertex in G' . Note that $N(v) \subseteq S$. Define $d(v)$ as red if $N(v)$ has only blue vertices. Furthermore, let $d(v) := \text{red}$ if $N(v)$ contains exactly one blue vertex. In all other cases, define $d(v) := \text{blue}$. This concludes the definition of d , it remains to show that d is indeed a CNCF-coloring.

Clearly, by this definition, for any $v \notin S$ we have that $N(v)$ is conflict-free colored by d , as we assumed that no such vertex had two red and two blue neighbors. It remains to show that for $v \in S$, $N(v)$ is conflict-free colored. Suppose for contradiction that it is not. Since any vertex $v \in S$ was conflict-free colored by c'' in G' , this implies that there exists a vertex $v \in S$ that has two red and two blue neighbors under d . Without loss of generality, suppose red was the conflict-free color of v in G' . Thus, there is a vertex $w \in V(G) \setminus V(G')$ that is a neighbor of v , with $d(w) := \text{red}$. But this contradicts that w is removed by the marking procedure, as we always keep at least two red neighbors of v if they exist. Thereby, d is a CNCF-coloring of G . \square

To obtain the kernel, for every set $X \subseteq S'$ of size at most two, mark 3 vertices $v \in V(G')$ with $X = N(v)$; if less than three such vertices exist, mark all. Remove all unmarked vertices from $V(G') \setminus S'$. This concludes the procedure. It follows from [11, Lemma 6] that this last step does not change the 2-CNCF-colorability of G' ; observe that this still holds after predefining the coloring of the vertex cover. It is easy to observe that $|S'| \leq 3|S|$ and $|V(G')| \leq |S'|^2 = \mathcal{O}(k^2)$. Furthermore, since any vertex in $G' \setminus S'$ has degree at most two, $|E(G')| \leq |S'|^2 + 2|V(G') \setminus S'| = \mathcal{O}(k^2)$. Using adjacency lists, this kernel can thus be stored in $\mathcal{O}(k^2 \log k)$ bits. \square

This completes the proof of Theorem 4.27.

5. Combinatorial bounds. Given a graph G , it is easy to prove that $\chi_{\text{CN}}(G) \leq \chi(G)$. However, there are examples that negate the existence of such bounds with respect to χ_{ON} [11]. In this section, we prove combinatorial bounds for χ_{ON} with respect to common graph parameters like treewidth, feedback vertex set, and vertex cover.

First, note that if G is a graph with isolated vertices then the graph can have no ONCF-coloring. Therefore, in all the arguments below we assume that G does not

have any isolated vertices. We obtain the following result. Recall that for a graph G , $\text{vc}(G)$, $\text{fvs}(G)$, and $\text{tw}(G)$ denote the size of a minimum vertex cover, the size of a minimum feedback vertex set, and the treewidth of G , respectively.

THEOREM 5.1. *Let G be a connected graph that has at least two vertices. The following statements hold:*

1. $\chi_{\text{ON}}(G) \leq 2\text{tw}(G) + 1$;
2. $\chi_{\text{ON}}(G) \leq \text{fvs}(G) + 3$;
3. $\chi_{\text{ON}}(G) \leq \text{vc}(G) + 1$. *Furthermore, if G is not a star graph or an edge-star graph, then $\chi_{\text{ON}}(G) \leq \text{vc}(G)$.*

In order to prove the above theorem, we prove each item separately. In the following lemma, we consider the bound on χ_{ON} with respect to the treewidth of a given graph.

LEMMA 5.2. *If G is a graph with treewidth t without isolated vertices, then it holds that $\chi_{\text{ON}}(G) \leq 2t + 1$.*

Proof. Consider a nice tree decomposition $\mathcal{T} = (T, \{X_{\mathbf{u}}\}_{\mathbf{u} \in V(T)})$ of G with root \mathbf{r} . We give a vertex coloring $c: V(G) \rightarrow [2t + 1]$ of G , which we will prove to be a $(2t + 1)$ -ONCF-coloring of G . Furthermore, we give a function $f: V(G) \rightarrow [2t + 1]$ such that $f(v)$ is the color that is uniquely used in the neighborhood of v . We will color the graph such that the following invariants hold after the handling of each bag.

- If two vertices are in the same bag $X_{\mathbf{i}}$ of \mathcal{T} that has already been handled, they receive distinct colors.
- The graph induced by the colored vertices is ONCF-colored, with the exception of vertices that are isolated in $G[Y]$, where Y is the set of vertices that has been colored thus far. Furthermore, the color $f(v)$ occurs exactly once in the coloring of $N(v)$.

Observe that by these two properties, c is not only a ONCF-coloring, but also a proper coloring of G .

Let $X_{\mathbf{r}}$ be the bag corresponding to the root \mathbf{r} , note that $|X_{\mathbf{r}}| \leq t + 1$. Color each of the vertices in $X_{\mathbf{r}}$ with a unique color from $[t + 1]$. For each vertex $v \in X_{\mathbf{r}}$, if $N(v) \cap X_{\mathbf{r}} \neq \emptyset$, pick one arbitrary vertex $u \in N(v) \cap X_{\mathbf{r}}$ and let $f(v) := c(u)$. Else, let $f(v) := 0$.

Below, we give a procedure to handle each of the remaining bags; we handle the bags of the tree decomposition in preorder. Let \mathbf{i} be any vertex of T such that its parent \mathbf{j} has been handled. We show how to color the vertices of $X_{\mathbf{i}}$. If $X_{\mathbf{i}} \subseteq X_{\mathbf{j}}$, all vertices of $X_{\mathbf{i}}$ have already been taken care of. Given that T is a nice tree decomposition, the only alternative is $X_{\mathbf{j}}$ is a forget node in T . Hence, $X_{\mathbf{i}} = X_{\mathbf{j}} \cup \{v\}$ for some $v \in V(G)$ and it only remains to color vertex v and possibly determine $f(u)$ for some vertices $u \in X_{\mathbf{i}}$.

Let $C := \{c(v) \mid v \in X_{\mathbf{j}}\}$ and let $F := \{f(v) \mid v \in X_{\mathbf{j}}\}$. Since $|X_{\mathbf{i}}| \leq t + 1$, it follows that $|X_{\mathbf{j}}| \leq t$. Thereby, $|F \cup C| \leq 2t$. We color v with a color from $[2t + 1] \setminus (F \cup C)$. If there is a vertex $u \in X_{\mathbf{j}}$ for which $f(u)$ was not yet set, and $\{u, v\} \in E(G)$, we let $f(u) = c(v)$. Furthermore, if v is not isolated in the graph constructed thus far, we let $f(v) = c(u)$ for some arbitrary $u \in X_{\mathbf{i}}$. Note that in the graph colored so far, $N(v) \subseteq X_{\mathbf{j}}$.

We show that these choices preserve the two invariants. The first invariant is immediate from the fact that v got a color distinct from the other vertices in its bag. All remaining vertices in $X_{\mathbf{i}}$ have distinct colors because the vertices in $X_{\mathbf{j}}$ have distinct colors.

To show the second variant, we need to show that $f(u)$ is set and occurs exactly once in $N(u)$, for all colored vertices u that have at least one colored neighbor. We consider three cases. (i) Let $u \neq v$ and $f(u)$ be set before the handling of X_i , then it is immediate that u has exactly one neighbor of color $f(u)$ in $N(u) \setminus \{v\}$. If v is a neighbor of u , then since u is already colored, it must be true that both u and v are present in X_i . Hence, we ensure that $c(v) \neq f(u)$, preserving the invariant. (ii) If $u \neq v$ and $f(u)$ was not yet set, then if $\{u, v\}$ is not an edge we do not set $f(u)$ as u is still isolated. If $\{u, v\}$ is an edge, then v is the only neighbor of u in the set of colored vertices, and we correctly set $f(u) := c(v)$. (iii) If $u = v$, we let $f(v)$ be the color of any of its neighbors in X_i . Since all neighbors of v that have been assigned a color must be present in X_i and hence in X_j , this implies they all have distinct colors, such that this choice is correct.

Since in the end, the entire graph is colored, the two invariants ensure that the result is an ONCF-coloring assuming that G has no isolated vertices. Since we use at most $2t + 1$ colors, $\chi_{\text{ON}}(G) \leq 2t + 1$. \square

A larger parameter than the treewidth of a graph is the size of a minimum feedback vertex set of a graph. In the following result, we compare χ_{ON} with the size of a minimum feedback vertex set.

LEMMA 5.3. *If G is a graph with a feedback vertex set (FVS) of size ℓ , then $\chi_{\text{ON}}(G) \leq \ell + 3$.*

Proof. Let X be an FVS of size ℓ in G . Using the color set $\mathcal{C} := \{r, g, b\} \cup \{c_x \mid x \in X\}$, we define a vertex coloring $c : V(G) \rightarrow \mathcal{C}$, which we prove is a $(\ell + 3)$ -ONCF-coloring of G . The general idea is as follows. Vertices in X are colored with c_x , and vertices not in X are conflict-free colored with $\{g, b\}$. The only problem with this coloring are vertices in $x \in X$ with no neighbors in X . We resolve this issue by a careful recoloring of some of these vertices with color r , and some of the vertices in $V(G) \setminus X$ with colors from $\{c_x \mid x \in X\}$.

Let $I := \{x \in X \mid N(x) \cap X = \emptyset\}$. Initialize $L = I$. Color the vertices in G by the following procedure.

1. For all $x \in X \setminus L$, let $c(x) := c_x$.
2. Each connected component Y of $G - X$ is a tree. If $|Y| > 1$, then ONCF-color Y with colors g and b ; this is possible by Observation 2.3. Otherwise (when $|Y| = 1$), color the single vertex in Y with g .
3. **while** there exists $u \in V(G) \setminus X$ such that $N(u) \subseteq L$
4. For each $x \in N(u)$ let $c(x) := c_x$
5. Let $L := L \setminus N(u)$
6. Pick an arbitrary $x \in N(u)$ and let $c(u) := c_x$
7. **for each** $x \in L$
8. Let $c(x) := r$
9. If there is no $y \in L$, $v \in N(x)$ such that $c(v) = c_y$
10. Pick an arbitrary $v \in N(x)$ and recolor v by letting $c(v) := c_x$

We argue that c is an ONCF-coloring. We consider the different vertices in G and argue their ONCF-coloring by c . Let L be the set L as in line 7 of the procedure.

- Consider a vertex $v \in V(G) \setminus X$ if $N(v) \setminus X \neq \emptyset$. In line 2, $N(v) \setminus X$ is given a conflict-free coloring using the colors $\{b, g\}$, which are not used for vertices in X . In line 9, some of these vertices may be recolored to c_x for $x \in L$, but observe that any such color is used at most once in G . This implies that $N(v)$ is conflict-free colored by c .

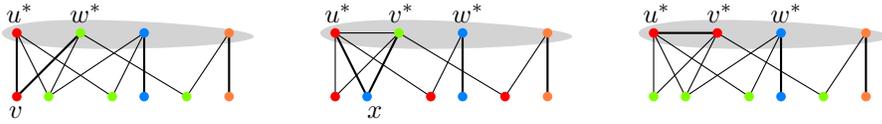


FIG. 7. (left) A coloring of the graph when all vertices in $G[S]$ are isolated. (middle) The case where $G[S]$ contains an edge and the endpoints have a common neighbor. (right) The case where $G[S]$ contains an edge and the endpoints have no common neighbors.

- For a vertex $v \in V(G) \setminus X$, if $N(v) \subseteq X$, then it cannot be the case that $N(v) \subseteq \{x \in X \mid c(x) = r\}$ as this would contradict the while loop in the definition of the coloring. All other colors are used at most once on vertices in X and therefore $N(v)$ is ONCF-colored by c .
- For a vertex $x \in X \setminus I$, there is a neighbor in X the color of which is unique in $V(G)$. Hence, $N(x)$ is ONCF-colored by c .
- Consider a vertex $x \in I$ with $c(x) = c_x$. Note that $N(x) \subseteq V(G) \setminus X$. By definition of c , there is a vertex $v \in V(G) \setminus X$ and $y \in I$ such that $x, y \in N(v) \subseteq I$ and $c(v) = c_y$. The color c_y is used at most once in $N(x)$, and thus it is ONCF-colored by c .
- Let $x \in I$ be a vertex with $c(x) = r$. Note that $N(x) \subseteq V(G) \setminus X$. By definition of c , there is a vertex $w \in N(x)$ that obtains a color c_y for some $y \in I$. Since color c_y is used only once in $V(G) \setminus X$, vertex w witnesses the ONCF-coloring of $N(x)$.

Thus, $\chi_{\text{ON}}(G) \leq \ell + 3$. □

Observe that the bound given in Lemma 5.3 is close to being tight. If we start from a clique K_k and subdivide each edge, the resulting graph has a feedback vertex set of size $k - 2$ and needs k colors to be ONCF-colored.

The next lemma bounds the value of $\chi_{\text{ON}}(G)$ for graphs with a vertex cover of size k . In particular, we improve the bound given by Gargano and Rescigno [11, Lemma 4], who showed that $\chi_{\text{ON}}(G) \leq 2k + 1$.

LEMMA 5.4. *Let G be a connected graph with $\text{vc}(G) = k$. Then $\chi_{\text{ON}}(G) \leq k + 1$. Furthermore, if G is not a star graph or an edge-star graph, then $\chi_{\text{ON}}(G) \leq k$.*

Proof. See Figure 7 for a sketch of the colorings described in the proof.

We start by proving the bounds for the case where G is not a star and not an edge-star. Let S be a minimum vertex cover of G and let k be the size of S . We do a case distinction on the size and connectedness of S .

($k = 2$ and S connected) First, we prove the bounds for $k = 2$ and $G[S]$ is an edge $\{u^*, v^*\}$. Note that G is not an edge-star graph. Therefore at least one of u^* or v^* have neighbors with degree exactly 1 in $G \setminus S$. We show that it is possible to ONCF-color such a graph with 2 colors, namely, r and b . Without loss of generality, let u^* have degree-1 neighbor w^* . We proceed as follows. Let $c(u^*) := c(w^*) := r$ and $c(v^*) := b$. For any other vertex in $V(G)$, let $c(v) := b$. It remains to verify that this is an ONCF-coloring. Any $v \notin S$ is clearly ONCF-colored by the fact that their neighborhood is a subset of S , and the vertices in S receive different colors. Furthermore, v^* has exactly one neighbor of color r (namely, u^*), and u^* has one neighbor of color r , namely w^* , concluding this part of the proof.

($G[S]$ disconnected or $k \geq 3$) We now prove the bounds for $k = 2$ and $G[S]$ is disconnected, and $k \geq 3$. We consider a number of cases.

(Suppose $G[S]$ contains a connected component C of size at least three.) Let $v^* \in C$ be a vertex such that $G[C \setminus \{v^*\}]$ remains connected. We color the vertices in G as follows. For every vertex $u \in S$, let $c(u) := c_u$. For every vertex $u \in S$ that is isolated in $G[S]$, pick an arbitrary neighbor $v \notin S$ and (re)color v such that $c(v) := c_u$. Notice that a vertex v in $G \setminus S$ may be picked multiple times as the candidate for an arbitrary neighbor for an isolated vertex in S , and in this case the color of this vertex v is set to the last color it is assigned. For every vertex v that is not yet colored, let $c(v) := c_{v^*}$.

Note that by this definition, every vertex in S has a distinct color. The colors that appear on vertices of $G \setminus S$ are either c_{v^*} or the color of a vertex that is isolated in $G[S]$. Also, by the choice of v^* , every vertex in the component C of $G[S]$ has at least one other neighbor in C . We verify that c is an ONCF-coloring.

- From the above, every vertex $u \in S$ that belongs to a connected component of size at least two in $G[S]$ has a uniquely colored neighbor in S that witnesses the ONCF-coloring of $N(u)$. Note that this includes v^* .
- Every isolated vertex u in $G[S]$ has $N(u) \subseteq V(G) \setminus S$. By description, every color c_v for $v \neq v^*$ occurs at most once in $V(G) \setminus S$, and u sees at least one such vertex. Thus, $N(u)$ is ONCF-colored.
- Finally, since all vertices in S are distinctly colored, the neighborhoods of vertices in $V(G) \setminus S$ are ONCF-colored.

Also, notice that the number of colors used is k . Thus, we are done in this case.

(Suppose $G[S]$ only contains connected components of size one.) Note that $|S| > 1$. Start by letting $c(v) := c_v$ for every vertex $v \in S$. Since G is connected, there exists $v \notin S$ such that $|N(v)| \geq 2$. Pick two vertices $u^*, w^* \in N(v)$ with $u^* \neq w^*$. Let $c(v) := c_{u^*}$. For every vertex $u \in S \setminus \{u^*, w^*\}$ pick an arbitrary neighbor $v \notin S$ and recolor v to c_u . Color the vertices that remained uncolored by this procedure with c_{w^*} . Note that by this procedure, every vertex in S has a distinct color. To see that c is an ONCF-coloring:

- Every vertex v in $G[S]$ has $N(v) \subseteq V(G) \setminus S$. By description, every color c_u for $u \neq w^*$ occurs at most once in $G \setminus S$. Furthermore, v has at least one neighbor with a color in $\{c_u \mid u \neq w^* \wedge u \in S\}$. Thus, $N(v)$ is ONCF-colored.
- Finally, since all vertices in S are distinctly colored, the neighborhoods of vertices in $V(G) \setminus S$ are ONCF-colored.

Also, notice that the number of colors used is k . Thus, we are done in this case.

(Otherwise.) In this case $G[S]$ has size at least 3, contains multiple connected components, and at least one such component has size two. Let $C = \{u^*, v^*\}$ be a connected component in $G[S]$ and let w^* be another arbitrarily chosen vertex. We do a further case distinction.

- Suppose there exists a vertex $x \notin S$ with $N(x) = \{u^*, v^*\}$; take one arbitrary such vertex. Then we let $c(v) := c_v$ for all $v \in S$, and we let $c(x) := c_{w^*}$. For every vertex $u \in S$ that is isolated in $G[S]$, pick an arbitrary neighbor $v \notin S$ and recolor v to $c(v) := c_u$. Define $c(v) := c_{u^*}$ for all vertices v that remained uncolored thus far. Notice that this coloring is very similar to the colorings in the previous cases. The only verification to be done is that for the sets $N(u^*)$ and $N(v^*)$ and both these neighborhoods have a vertex x that is uniquely colored with c_{w^*} . Thus, with arguments similar to those in the previous cases we obtain a k -ONCF-coloring for G in this case.
- Alternatively, if there exists no vertex x with $N(x) = \{u^*, v^*\}$, then we let $c(u^*) = c(v^*) = c_{w^*}$, and we let $c(v) := c_v$ for all vertices in $S \setminus \{u^*, v^*\}$. For every vertex $u \in S$ that is isolated in $G[S]$, pick an arbitrary neighbor

$v \in N(u)$ and recolor v to $c(v) := c_u$. Color all remaining vertices with c_{v^*} . Notice that $N(u^*)$ and $N(v^*)$ have v^* and u^* uniquely colored with c_{u^*} , respectively. Using arguments similar to previous cases, we can show that the described coloring is a k -ONCF-coloring.

If G is not a star and not an edge-star, we are in one of the cases above. Otherwise, it is easy to observe that stars have a vertex cover of size one and can always be colored with two colors, and edge-stars can be colored with three colors while having a minimum vertex cover size of two. \square

Observe that the bounds of Lemma 5.4 are tight. First, a star graph requires 2 colors and has vertex cover size 1 while an edge-star graph requires 3 colors and has vertex cover size 2. On the other hand, given a $q \geq 3$, taking the complete graph K_q and subdividing each edge once results in a graph that requires q colors [11] for an ONCF-coloring and has a vertex cover of size q .

Using Lemmas 5.2, 5.3, and 5.4 we complete the proof of Theorem 5.1.

6. Open problems. The study in this paper leads to some interesting open questions. In this paper we only exhibit a generalized kernel of size $\mathcal{O}(k^9)$ for 2-CNCF-COLORING and it remains to resolve the size of tight polynomial kernels for the problem. On the combinatorial side, with respect to minimum vertex cover, we obtain tight upper bounds on $\chi_{ON}(G)$. It would be interesting to obtain corresponding tight bounds for $\chi_{ON}(G)$ with respect to FVS and treewidth.

REFERENCES

- [1] Z. ABEL, V. ALVAREZ, E. D. DEMAINE, S. P. FEKETE, A. GOUR, A. HESTERBERG, P. KELDENICH, AND C. SCHEFFER, *Conflict-free coloring of graphs*, SIAM J. Discrete Math., 32 (2018), pp. 2675–2702, <https://doi.org/10.1137/17M1146579>.
- [2] D. AJWANI, K. ELBASSIONI, S. GOVINDARAJAN, AND S. RAY, *Conflict-free coloring for rectangle ranges using $O(n^{382})$ colors*, Discrete Comput. Geom., 48 (2012), pp. 39–52, <https://doi.org/10.1007/s00454-012-9425-5>.
- [3] H. L. BODLAENDER, B. M. P. JANSEN, AND S. KRATSCH, *Kernelization lower bounds by cross-composition*, SIAM J. Discrete Math., 28 (2014), pp. 277–305, <https://doi.org/10.1137/120880240>.
- [4] H. L. BODLAENDER, S. THOMASSÉ, AND A. YEO, *Kernel bounds for disjoint cycles and disjoint paths*, Theoret. Comput. Sci., 412 (2011), pp. 4570–4578, <https://doi.org/10.1016/j.tcs.2011.04.039>.
- [5] P. CHEILARIS, *Conflict-free Coloring*, PhD thesis, City University of New York, New York, 2009.
- [6] M. CYGAN, F. V. FOMIN, L. KOWALIK, D. LOKSHTANOV, D. MARX, M. PILIPCZUK, M. PILIPCZUK, AND S. SAURABH, *Parameterized Algorithms*, Springer, Cham, Switzerland, 2015, <https://doi.org/10.1007/978-3-319-21275-3>.
- [7] A. DARMANN AND J. DÖCKER, *On a simple hard variant of not-all-equal 3-sat*, Theoret. Comput. Sci., 815 (2020), pp. 147–152, <https://doi.org/https://doi.org/10.1016/j.tcs.2020.02.010>.
- [8] G. EVEN, Z. LOTKER, D. RON, AND S. SMORODINSKY, *Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks*, SIAM J. Comput., 33 (2003), pp. 94–136, <https://doi.org/10.1137/S0097539702431840>.
- [9] F. V. FOMIN AND D. KRATSCH, *Exact Exponential Algorithms*, Springer, Berlin, 2010.
- [10] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability; A Guide to the Theory of NP-Completeness*, Freeman, New York, 1990.
- [11] L. GARGANO AND A. A. RESCIGNO, *Complexity of conflict-free colorings of graphs*, Theoret. Comput. Sci., 566 (2015), pp. 39–49, <https://doi.org/10.1016/j.tcs.2014.11.029>.
- [12] R. GLEBOV, T. SZABÓ, AND G. TARDOS, *Conflict-free colouring of graphs*, Combin. Probab. Comput., 23 (2014), pp. 434–448.
- [13] S. HAR-PELED AND S. SMORODINSKY, *Conflict-free coloring of points and simple regions in the plane*, Discrete Comput. Geom., 34 (2005), pp. 47–70, <https://doi.org/10.1007/s00454-005-1162-6>.

- [14] B. M. P. JANSEN AND S. KRATSCH, *Data reduction for graph coloring problems*, Inform. and Comput., 231 (2013), pp. 70–88, <https://doi.org/10.1016/j.ic.2013.08.005>.
- [15] B. M. P. JANSEN AND A. PIETERSE, *Optimal data reduction for graph coloring using low-degree polynomials*, Algorithmica, 81 (2019), pp. 3865–3889.
- [16] B. M. P. JANSEN AND A. PIETERSE, *Optimal sparsification for some binary CSPs using low-degree polynomials*, ACM Trans. Comput. Theory, 11 (2019), 28.
- [17] D. LOKSHANOV, D. MARX, AND S. SAURABH, *Known algorithms on graphs of bounded treewidth are probably optimal*, ACM Trans. Algorithms, 14 (2018), 13, <https://doi.org/10.1145/3170442>.
- [18] J. PACH AND G. TARDOS, *Conflict-free colourings of graphs and hypergraphs*, Comb. Probab. Comput., 18 (2009), pp. 819–834, <https://doi.org/10.1017/S0963548309990290>.
- [19] T. J. SCHAEFER, *The complexity of satisfiability problems*, in Proc. 10th STOC, ACM, New York, 1978, pp. 216–226, <https://doi.org/10.1145/800133.804350>.
- [20] S. SMORODINSKY, *Combinatorial Problems in Computational Geometry*, PhD thesis, School of Computer Science, Tel Aviv University, Tel Aviv, Israel, 2003.
- [21] S. SMORODINSKY, *Conflict-Free Coloring and its Applications*, Springer, Berlin, 2013.