

Clique-width III: Hamiltonian Cycle and the Odd Case of Graph Coloring

FEDOR V. FOMIN, PETR A. GOLOVACH, and DANIEL LOKSHTANOV,

Department of Informatics, University of Bergen, Norway

SAKET SAURABH, Department of Informatics, University of Bergen, Norway and The Institute of Mathematical Sciences, HBNI, India

MEIRAV ZEHAVID, Computer Science Department, Ben-Gurion University of the Negev, Israel

MAX-CUT, EDGE DOMINATING SET, GRAPH COLORING, and HAMILTONIAN CYCLE on graphs of bounded clique-width have received significant attention as they can be formulated in MSO_2 (and, therefore, have linear-time algorithms on bounded treewidth graphs by the celebrated Courcelle's theorem), but cannot be formulated in MSO_1 (which would have yielded linear-time algorithms on bounded clique-width graphs by a well-known theorem of Courcelle, Makowsky, and Rotics). Each of these problems can be solved in time $g(k)n^{f(k)}$ on graphs of clique-width k . Fomin et al. (2010) showed that the running times cannot be improved to $g(k)n^{O(1)}$ assuming $\text{W}[1] \neq \text{FPT}$. However, this does not rule out non-trivial improvements to the exponent $f(k)$ in the running times. In a follow-up paper, Fomin et al. (2014) improved the running times for EDGE DOMINATING SET and MAX-CUT to $n^{O(k)}$, and proved that these problems cannot be solved in time $g(k)n^{o(k)}$ unless ETH fails. Thus, prior to this work, EDGE DOMINATING SET and MAX-CUT were known to have tight $n^{\Theta(k)}$ algorithmic upper and lower bounds.

In this article, we provide lower bounds for HAMILTONIAN CYCLE and GRAPH COLORING. For HAMILTONIAN CYCLE, our lower bound $g(k)n^{o(k)}$ matches asymptotically the recent upper bound $n^{O(k)}$ due to Bergougnoux, Kanté, and Kwon (2017).

As opposed to the asymptotically tight $n^{\Theta(k)}$ bounds for EDGE DOMINATING SET, MAX-CUT, and HAMILTONIAN CYCLE, the GRAPH COLORING problem has an upper bound of $n^{O(2^k)}$ and a lower bound of merely $n^{o(\sqrt{k})}$ (implicit from the $\text{W}[1]$ -hardness proof). In this article, we close the gap for GRAPH COLORING by proving a lower bound of $n^{2^{o(k)}}$. This shows that GRAPH COLORING behaves qualitatively different from the other three problems. To the best of our knowledge, GRAPH COLORING is the first natural problem known to require exponential dependence on the parameter in the exponent of n .

CCS Concepts: • **Mathematics of computing** → **Graph coloring; Graph algorithms**; • **Theory of computation** → **Parameterized complexity and exact algorithms**;

The preliminary version of this article appeared as an extended abstract in the proceedings of SODA 2018.

This work is supported by the Research Council of Norway via the projects "CLASSIS" and "MULTIVAL", the Israel Science Foundation under Grant No. 1176/18, and the European Research Council under ERC Starting Grant No. 715744 "Pareto-Optimal Parameterized Algorithms".

Authors' addresses: F. V. Fomin, P. A. Golovach, and D. Lokshtanov, Department of Informatics, University of Bergen, Bergen, PB 7803, 5020, Norway; emails: {fedor.fomin, petr.golovach}@uib.no, daniello@ii.uib.no; S. Saurabh, The Institute of Mathematical Sciences, HBNI, 4th Cross Street, CIT Campus, Tharamani, Chennai, Tamil Nadu, 600113, India; email: saket@imsc.res.in; M. Zehavi, Computer Science Department, Ben-Gurion University of the Negev, Alon High-Tech Building, Beersheba, Israel; email: meiravze@bgu.ac.il.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

1549-6325/2018/11-ART9 \$15.00

<https://doi.org/10.1145/3280824>

Additional Key Words and Phrases: Coloring, Hamiltonian cycle, fine-grained complexity, exponential time hypothesis

ACM Reference format:

Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. 2018. Clique-width III: Hamiltonian Cycle and the Odd Case of Graph Coloring. *ACM Trans. Algorithms* 15, 1, Article 9 (November 2018), 27 pages.

<https://doi.org/10.1145/3280824>

1 INTRODUCTION

Many NP-hard problems become polynomial time solvable on trees and cliques. This has motivated researchers to look for families of graphs that have algorithmic properties similar to those of trees and cliques. In particular, ideas of being “tree-like” and “clique-like” were explored, leading to the notions of *treewidth* and *clique-width*, respectively. Treewidth has been introduced independently by several authors over the last 50 years. It was first introduced by Bertelé and Brioschi [3] in 1972 under the name of *dimension*. Later, it was rediscovered by Halin [30], and finally, in 1984, Robertson and Seymour [44] introduced it under the current name, as a part of their Graph Minors project. Since then, the notion of treewidth has been studied by several authors, and now it is one of the most important parameters in graph algorithms. We refer to the survey of Bodlaender [4] for further references on treewidth.

The notion of treewidth captures the fact that trees are structurally simple, but fails to do this for cliques. In fact, the treewidth of a clique on n vertices is $n - 1$. Courcelle and Olariu [15] defined new kind of graph decompositions that capture the structure both of bounded treewidth graphs and of cliques and clique-like graphs, and at the same time enjoy most of the algorithmic properties of bounded treewidth graphs. The corresponding notion that measures the quality of the decomposition was called the *clique-width* of the graph. Clique-width is a generalization of treewidth in the sense that graphs of bounded treewidth also have bounded clique-width [12]. It is also worth mentioning here the related graph parameters NLC-width, introduced by Wanke [46], rankwidth introduced by Seymour and Oum [41], and Booleanwidth, introduced by Bui-Xuan, Telle, and Vatshelle [8]. We refer to the survey of Hliněný et al. [32] for further references on clique-width and related parameters.

In the last decade, clique-width as a graph parameter has received significant attention. Corneil et al. [11] show that graphs of clique-width at most three can be recognized in polynomial time. Fellows et al. [22] settled a long-standing open problem by showing that computing clique-width is NP-hard. Oum and Seymour [41] describe an algorithm that, for any fixed k , runs in time $O(n^9 \log n)$ and computes $(2^{3k+2} - 1)$ -expressions for a n -vertex graph G of clique-width at most k .¹ Oum [42] improved this result by providing an algorithm computing $(8^k - 1)$ -expressions in time $O(n^3)$. Finally, Hliněný and Oum [31] obtained an algorithm running in time $O(n^3)$ and computing $(2^{k+1} - 1)$ -expressions for a graph G of clique-width at most k .

Most of the algorithms on graphs of bounded treewidth or clique-width are based on dynamic programming over the corresponding decomposition tree and are very similar to each other. This similarity hinted at the existence of meta-theorems that could simultaneously provide algorithms on bounded treewidth and clique-width graphs for large classes of problems. Indeed, Courcelle [13] (see, also, Ref. [1]) proved that every problem expressible in monadic second-order logic (MSO_2), say, by a sentence ϕ , is solvable in time $f(|\phi|, k) \cdot n$ on graphs with n vertices and treewidth k .

¹The clique-width of a graph is the minimum t for which it admits a decomposition of with t called a t -expression, defined in Section 2.

That is, these problems are fixed parameter tractable (FPT) parameterized by the treewidth and the length of the formula. For problems expressible in monadic second-order logic with logical formulas that do not use edge set quantifications (so-called MSO_1), Courcelle, Makowsky, and Rotics [14] extended the meta-theorem of Courcelle to graphs of bounded clique-width. More concretely, they proved that every problem expressible in MSO_1 , say, by a sentence ϕ , is solvable in time $\tau(|\phi|, k) \cdot n$ on graphs with n vertices and clique-width k . Thus, these problems are FPT parameterized by the clique-width and the length of the formula.

Comparing the two meta-theorems reveals a tradeoff between expressive power of the logic and applicability to larger (bounded cliquewidth) or smaller (bounded treewidth) classes of graphs. This leads to the question on whether this tradeoff is unavoidable. Courcelle, Makowsky, and Rotics [14] addressed this question and proved that there exist problems that are definable in MSO_2 but are not polynomial time solvable, even on cliques, unless $\text{NEXP} = \text{EXP}$. For several natural graph problems, such as MAX-CUT , $\text{EDGE DOMINATING SET}$, GRAPH COLORING , and HAMILTONIAN CYCLE , linear time algorithms on bounded treewidth graphs were known to follow from (variants of [1, 6]) Courcelle's theorem [13]. At the same time, these problems, and many others, were known to admit algorithms with running time $O(n^{f(k)})$ on graphs with n vertices and clique-width k [21, 26–28, 35, 36, 39, 43, 45, 46].

The existence of FPT algorithms (parameterized by the clique-width k of the input graph) for these problems (or their generalizations) was asked as open problems by Gerber and Kobler [26]; Kobler and Rotics [35, 36]; and, Makowsky, Rotics, Averbouch, Kotek, and Godlin [28, 39].

A subset of the authors in Ref. [23] showed that the EDS , HC , and GC problems parameterized by clique-width are all $\text{W}[1]$ -hard. In particular, this implies that these problems do not admit algorithms with running times of the form $O(g(k) \cdot n^c)$, for any function g and constant c independent of k , unless $\text{FPT}=\text{W}[1]$. However, the lower bounds of Fomin et al. [23] did not rule out non-trivial improvements to the exponent $f(k)$ of n in the running times.

In a follow-up article, Fomin et al. [24] improved the running times for $\text{EDGE DOMINATING SET}$ and MAX-CUT from $n^{O(k^2)}$ to $n^{O(k)}$, and proved $g(k)n^{o(k)}$ lower bounds for $\text{EDGE DOMINATING SET}$ and MAX-CUT , assuming the Exponential Time Hypothesis (ETH). Together, these lower and upper bounds gave asymptotically tight algorithmic bounds for $\text{EDGE DOMINATING SET}$ and MAX-CUT . However, for HAMILTONIAN CYCLE and GRAPH COLORING , large gaps remained between the known running time upper and lower bounds. This article bridges the gaps for GRAPH COLORING and HAMILTONIAN CYCLE by proving new lower bounds, which asymptotically match the known upper bounds.

GRAPH COLORING has an upper bound of $n^{O(2^k)}$ [36] and, prior to this work, a lower bound of merely $n^{o(\sqrt{k})}$ (implicit from the $\text{W}[1]$ -harness proof). Our first theorem shows that the upper bound is asymptotically tight, by providing a lower bound of $n^{2^{o(k)}}$. Specifically, we prove the following.

THEOREM 1. *Unless ETH fails, GRAPH COLORING cannot be solved in time $f(k) \cdot n^{2^{o(k)}}$ for any function f of k , where k is the clique-width of the input graph.*

In fact, we prove a stronger result, and Theorem 1 follows as a corollary. Specifically, we prove that the lower bound of Theorem 1 holds even for graphs of *linear clique-width* k , when a linear clique-width expression (see Ref. [29]) of width at most k is given as input.

Theorem 1 shows that GRAPH COLORING behaves qualitatively different from every other problem previously studied on graphs of bounded clique-width. Indeed, to the best of our knowledge, GRAPH COLORING parameterized by clique-width is the first (natural) parameterized problem known to require exponential dependence on the parameter in the exponent of n . Note here that

there do exist problems for which the tight upper and lower bounds on the dependence of the running time on the parameter are double exponential, triple exponential, or even non-elementary (see, e.g., Refs [17, 25, 37, 40]). However, these lower bounds are all for the $g(k)$ factor of FPT algorithms and not for the exponent of the input size n .

Our second theorem provides a lower bound $g(k) \cdot n^{o(k)}$ for HAMILTONIAN CYCLE, where k is the clique-width of the input graph. This result was announced (without a proof) in the concluding section of Ref. [24]. At the time of publishing Ref. [24], the best known upper bound for HAMILTONIAN CYCLE was $n^{O(k^2)}$. Due to the significant gap between the known lower and upper bounds for HAMILTONIAN CYCLE, the proof of the $g(k) \cdot n^{o(k)}$ lower bound for HAMILTONIAN CYCLE was omitted from Ref. [24]. In 2017, Bergougnoux, Kanté, and Kwon [2] closed this gap by providing a beautiful new algorithm with running time $n^{O(k)}$. In other words, Bergougnoux, Kanté, and Kwon [2] showed that the $g(k) \cdot n^{o(k)}$ lower bound for HAMILTONIAN CYCLE claimed in Ref. [24] is tight. In this article, we provide a full proof of this claim. In particular, we prove the following.

THEOREM 2. *Unless ETH fails, HAMILTONIAN CYCLE cannot be solved in time $f(k) \cdot n^{o(k)}$ for any function f of k , where k is the clique-width of the input graph.*

Overview. The remaining part of the article is organized as follows. In Section 2, we set up basic notations and definitions. Section 3 is devoted to the proof of Theorem 1 and it has the following structure. In Section 3.1, we define an intermediate problem called 4-MONOTONE MIN-CSP and prove a running time lower bound for this problem. The proof of this lower bound (Section 3.1) is quite standard and can be skipped by a reader interested in going directly to the crux of our lower bound proof—the reduction from 4-MONOTONE MIN-CSP to GRAPH COLORING on graphs of bounded clique-width. This reduction is presented in Section 3.2. We remark, however, that our intermediate problem can potentially help to obtain other lower bounds. In Section 4, we prove Theorem 2 about HAMILTONIAN CYCLE. In Section 5, we wrap up with concluding remarks and open problems.

2 PRELIMINARIES

We use $[n]$ and $[n]_0$ as shorthands for $\{1, 2, \dots, n\}$ and $\{0, 1, \dots, n\}$, respectively. Given a function $f : A \rightarrow B$, we let $\text{dom}(f)$ and $\text{ima}(f)$ denote the domain and image of f , respectively. Moreover, given $A' \subseteq A$, we denote $f(A') = \{f(a) : a \in A'\}$.

Basic Graph Theory. We refer to standard terminology from the book of Diestel [18] for those graph-related terms that are not explicitly defined here. Given a graph G , we denote its vertex set and its edge set by $V(G)$ and $E(G)$, respectively. Moreover, when the graph G is clear from context, denote $n = |V(G)|$. Given a subset $U \subseteq V(G)$, $G[U]$ denotes the subgraph of G induced by U . For $X \subseteq V(G)$, $G - X$ denotes the graph obtained from G by the deletion of the vertices of X , i.e., $G - X = G[V(G) \setminus X]$. We say that G is a *clique* if for all distinct vertices $u, v \in V(G)$, we have that $\{u, v\} \in E(G)$, and that $V(G)$ is an independent set if for all distinct vertices $u, v \in V(G)$, we have that $\{u, v\} \notin E(G)$. Given a vertex $v \in V(G)$, $N_G(v)$ denotes the neighborhood of v in G . Moreover, given two subsets $U, T \subseteq V(G)$, the subset U is a *module with respect to T* if for all $u, u' \in U$ and $v \in T$, either both u and u' are adjacent to v or both u and u' are not adjacent to v , and if, in addition, $T = V(G) \setminus U$, then U is simply called a *module*. A *matching* M in G is a subset of $E(G)$ whose edges do not share any endpoint, and a *perfect matching* M is a matching of size $n/2$ (that is, every vertex in $V(G)$ is incident to exactly one edge in M). A *feedback vertex set* of a graph is a set of vertices $X \subseteq V(G)$ such that $G - X$ is a forest. The *feedback vertex number* of a graph G , denoted as $\text{fvn}(G)$, is the minimum size of a feedback vertex set of G .

A *coloring* of G is a function $\chi : V(G) \rightarrow \mathbb{N}$. The integers in the codomain of χ are called *colors*. We say that χ is a *proper coloring* of G if for every edge $\{u, v\} \in E(G)$, we have that $\chi(u) \neq \chi(v)$. Moreover, a subgraph H of G is said to be *multicolored* if for all distinct vertices $u, v \in V(H)$, we have that $\chi(u) \neq \chi(v)$. We remark that a clique is multicolored if and only if it is properly colored. The *chromatic number* of G is the smallest integer t such that G has a proper coloring $\chi : V(G) \rightarrow [t]$, that is, a proper coloring that uses only t colors.

A cycle C of a graph G is *Hamiltonian* if C contains all the vertices of G . Respectively, G is said to be *Hamiltonian* if it has a Hamiltonian cycle.

Clique-width. Let G be a graph, and t be a positive integer. A t -*graph* is a graph with vertices labeled by integers from $[t]$. We refer to a t -graph consisting of exactly one vertex labeled by some integer from $[t]$ as to an *initial t -graph*. The *clique-width* $\text{cw}(G)$ of G is the smallest integer t such that G can be constructed by means of repeated application of the following four operations:

- $i(v)$: *Introduce* operation constructing an initial t -graph with vertex v labeled by i ,
- \oplus : *Disjoint union*,
- $\rho_{i \rightarrow j}, i \neq j$: *Relabel* operation changing all labels i to j , and
- $\eta_{i,j}, i \neq j$: *Join* operation making all vertices labeled by i adjacent to all vertices labeled by j .

Respectively, an *expression tree* of a graph G defined as a rooted tree T with nodes of four types i, \oplus, η and ρ :

- *Introduce nodes* $i(v)$ are leaves of T corresponding to initial t -graphs with vertices v labeled by i .
- *Union node* \oplus stands for a disjoint union of graphs associated with its children.
- *Relabel node* $\rho_{i \rightarrow j}$ has one child and is associated with the t -graph obtained by applying of the relabeling operation to the graph corresponding to its child.
- *Join node* $\eta_{i,j}$ has one child and is associated with the t -graph resulting by applying the join operation to the graph corresponding to its child.
- The graph G is isomorphic to the graph associated with the root of T (with all labels removed).

The *width* of the tree T is the number of different labels appearing in T . We have that $\text{cw}(G) = t$ if and only if there is a rooted expression tree T of width t of G . We call the elements of $V(T)$ *nodes* to distinguish them from the vertices of G . Given a node X of an expression tree of G , the graph G_X represents the graph formed by the subtree T_X of the expression tree rooted at X .

The *linear clique-width* $\text{lcw}(G)$ of G is defined similarly, except that now the application of the operation \oplus is restricted as follows: for two t -graphs G_1 and G_2 , we can perform the operation $G_1 \oplus G_2$ only if at least one graph among G_1 and G_2 is an initial t -graph. Clearly, as the set of operations relevant to linear clique-width is more restrictive than the set of operations relevant to clique-width, the following observation is correct.

OBSERVATION 2.1. *For any graph G , $\text{cw}(G) \leq \text{lcw}(G)$.*

Let us now present an almost equivalent definition of linear clique-width, known as the *neighborhood width* [29]. To this end, let σ be an ordering of $V(G)$ as $v_1^\sigma, v_2^\sigma, \dots, v_n^\sigma$. For all $i \in [n]$, denote $V_i^\sigma = \{v_1^\sigma, v_2^\sigma, \dots, v_i^\sigma\}$. Two vertices $u, v \in V_i^\sigma$ are *i -equivalent under σ* if their neighborhoods outside V_i^σ are identical, that is, $N_G(u) \setminus V_i^\sigma = N_G(v) \setminus V_i^\sigma$. Accordingly, the *i -equivalence partition under σ* , denoted by $\text{EQ}(G, \sigma, i)$, is a partition $\{S_1, S_2, \dots, S_t\}$ of V_i^σ , for some $t \in [n]$, that satisfies **(i)** for all $j \in [t]$, every two vertices $u, v \in S_j$ are i -equivalent under σ , and **(ii)** for all $j, \ell \in [t]$, every $u \in S_j$ and $v \in S_\ell$ are not i -equivalent under σ . As the notion of i -equivalence

under σ defines an equivalence relation, this partition is well-defined. Specifically, $EQ(G, \sigma, i)$ is the partition of V_i^σ into the equivalence classes of the relation i -equivalent under σ .

Definition 2.1. Let G be a graph. For an ordering σ of G , the *neighborhood-width of G under σ* is defined as $nw(G, \sigma) \triangleq \max_{i \in [n]} |EQ(G, \sigma, i)|$. Furthermore, the *neighborhood-width of G* is defined as $nw(G) = \min_{\sigma} nw(G, \sigma)$ where σ ranges over all possible orderings of $V(G)$.

The following proposition asserts that, for our purpose, we can work with $nw(G)$ rather than $1cw(G)$.

PROPOSITION 2.1 ([29]). *For any graph G , $1cw(G) \leq nw(G) + 1$.*

Parameterized Complexity. Let Π be an NP-hard problem. In the framework of Parameterized Complexity, each instance of Π is associated with a *parameter* k . Here, the goal is to confine the combinatorial explosion in the running time of an algorithm for Π to depend only on k . Formally, we say that Π is FPT if any instance (I, k) of Π is solvable in time $f(k) \cdot |I|^{O(1)}$, where f is an arbitrary function of k . A weaker request is that for every fixed k , the problem Π would be solvable in polynomial time. Formally, we say that Π is *slice-wise polynomial (XP)* if any instance (I, k) of Π is solvable in time $f(k) \cdot |I|^{g(k)}$, where f and g are arbitrary functions of k . Nowadays, Parameterized Complexity supplies a rich toolkit to design FPT and XP algorithms, or to show that such algorithms are unlikely to exist.

To obtain (essentially) tight conditional lower bounds for the running time of FPT or XP algorithms, we rely on the well-known ETH [9, 33, 34]. To formalize the statement of ETH, we first recall that given a formula φ in conjunctive normal form (CNF) with n variables and m clauses, the task of CNF-SAT is to decide whether there is a truth assignment to the variables that satisfies φ . In the p -CNF-SAT problem, each clause is restricted to have at most p literals. ETH asserts that 3-CNF-SAT cannot be solved in time $O(2^{o(n)})$. Additional details on Parameterized Complexity and ETH can be found in Refs. [16, 20].

3 GRAPH COLORING

In this section, we prove Theorem 1. The proof is quite involved, and before diving into technical details, we provide some intuition about how it goes.

The key insights of the proof are in some sense dual to the key insights of the $n^{2^{O(k)}}$ time algorithm [36]. It is convenient to consider graphs of bounded *neighborhood-width* rather than bounded clique-width. In this setting, the vertices of G are given according to an ordering $\sigma = v_1^\sigma, v_2^\sigma, \dots, v_n^\sigma$, and satisfy the following property. For every $i \leq n$, the vertex set $\{v_1^\sigma, \dots, v_i^\sigma\}$ can be partitioned into k sets S_1, \dots, S_k such that the sets S_j are “equivalence classes with respect to the future” in the following sense. For every set S_j , all of the vertices in S_j have exactly the same neighborhood in $\{v_{i+1}^\sigma, \dots, v_n^\sigma\}$.

Consider a coloring algorithm that tries to color the vertices of G in the order given by σ using at most η colors. When the vertices $\{v_1^\sigma, \dots, v_i^\sigma\}$ have already been colored, this affects which colors can be used on the remaining vertices. For each color c , the set of vertices in $\{v_{i+1}^\sigma, \dots, v_n^\sigma\}$ that *cannot* be colored by c are exactly the vertices that have at least one neighbor in $\{v_1^\sigma, \dots, v_i^\sigma\}$ colored with c . This vertex set is completely determined by the subset I_c of $\{1, \dots, k\}$ of indices such that $j \in I_c$ if and only if some vertex in S_j has been colored with c . In other words, two color classes c and c' for which I_c and $I_{c'}$ are the same are interchangeable—any vertex in $\{v_{i+1}^\sigma, \dots, v_n^\sigma\}$ that can be colored with c can be colored with c' instead and vice versa. Hence, to completely describe how the partial coloring affects what can be done in the future, it is sufficient to record, for every subset I of $\{1, \dots, k\}$, the number of colors c such that $I_c = I$. This gives rise to an $n^{2^{O(k)}}$ time dynamic programming algorithm.

To prove the lower bound, we encode instances of the “ 2^k -CLIQUE” problem in terms of graph coloring on graphs of neighborhood-width $\mathcal{O}(k)$. In the 2^k -CLIQUE problem, the input is a graph G on n vertices, an integer k , and the task is to determine whether the graph contains a clique of size 2^k . Since the usual k -CLIQUE problem can not be solved in time $f(k)n^{o(k)}$ [10, 16] assuming the ETH, the 2^k -CLIQUE problem can not be solved in time $f(k)n^{o(2^k)}$ under the same assumption.

In the 2^k -CLIQUE problem, one has to select 2^k vertices correctly out of a set of n candidates. There is a natural correspondence between selecting “one out of n vertices” in the 2^k -CLIQUE and selecting one number n_I between $1, \dots, n$ —for a fixed subset I the number of colors c such that $I_c = I$. In other words, the selection of a vertex is encoded as the number of color classes of a specific “type,” where the type of a color is which of the sets S_1, \dots, S_k it intersects. While the correspondence itself is natural, carrying out the reduction is a rather delicate task. In particular, it is challenging to “implement” vertex selection in terms of selecting the numbers n_I , and “implementing” adjacency testing only using relations between the numbers n_I , without, at the same time, increasing the neighborhood width too much. The crucial gadget used to achieve this is the “Mini-Constraint Selector” introduced in Section 3.2.

3.1 Reduction to MONOTONE MIN-CSP

The starting point of our proof of Theorem 1 is the MULTICOLORED CLIQUE problem, which is defined as follows.

MULTICOLORED CLIQUE (Parameterized by Solution Size)	Parameter: k
Input: A graph G with a coloring $\chi : V(G) \rightarrow [k]$.	
Question: Does G contain a multicolored clique C on k vertices?	

For MULTICOLORED CLIQUE, we have the following known proposition.

PROPOSITION 3.1 ([38]). *Unless ETH fails, MULTICOLORED CLIQUE cannot be solved in time $f(k) \cdot n^{o(k)}$ for any function f of k .*

The focus of this section is to reduce MULTICOLORED CLIQUE to a new problem that we call MONOTONE MIN-CSP. Later, in Section 3.2, we present the main part of our proof, which is a reduction from MONOTONE MIN-CSP to GRAPH COLORING. Let us first formally define the MONOTONE MIN-CSP problem. To this end, let X be a set of variables whose size is denoted by k . Let $n \in \mathbb{N}$. A function $\alpha : X \rightarrow [n]_0$ is called an *assignment*. The *cost* of an assignment α , denoted by $\text{cost}(\alpha)$, is $\sum_{x \in X} \alpha(x)$. Given $X' \subseteq X$, a set R of pairs (x, c) such that $x \in X'$ and $c \in [n]_0$ is called an X' -*mini-constraint*, or simply a *mini-constraint*. We say that an assignment α satisfies a mini-constraint R if for all $(x, c) \in R$, we have that $\alpha(x) \geq c$. A *constraint* is a pair $C = (X', \mathcal{R})$, where $X' \subseteq X$ and \mathcal{R} is a set of X' -mini-constraints. The *arity* of a constraint $C = (X', \mathcal{R})$ is $|X'|$. We say that an assignment α satisfies a constraint $C = (X', \mathcal{R})$ if α satisfies *at least one* mini-constraint $R \in \mathcal{R}$. Furthermore, we say that an assignment α satisfies a set C of constraints if α satisfies *every* constraint in C .

MONOTONE MIN-CSP (Parameterized by Variable Number)	Parameter: $ X = k$
Input: A set of variables X , a set of constraints C and $n, W \in \mathbb{N}$.	
Question: Does there exist an assignment of cost at most W that satisfies C ?	

The special case of MONOTONE MIN-CSP where the arity of every input constraint is at most r , for some fixed $r \in \mathbb{N}$, is called r -MONOTONE MIN-CSP. The rest of this section is devoted to the proof of the following lemma.

LEMMA 3.1. *Unless ETH fails, 4-MONOTONE MIN-CSP cannot be solved in time $f(k) \cdot n^{o(k)}$ for any function f of k .*

Construction. Let (G, χ, k) be an instance of MULTICOLORED CLIQUE. Without loss of generality, we assume that for all $i, j \in [k]$, it holds that $|\chi^{-1}(i)| = |\chi^{-1}(j)|$, and denote this size by n' . Indeed, this condition can be easily ensured by adding isolated vertices of the appropriate colors to G . For every color $i \in [k]$, we denote $\chi^{-1}(i) = \{v_1^i, v_2^i, \dots, v_{n'}^i\}$.

Let us now construct an instance $\text{red}(G, \chi, k) = (X, C, n', W)$ of 4-MONOTONE MIN-CSP, where n' is as defined above and $|X| \triangleq k' = 2k$ (the value k is the same in both instances). First, we define $X = \{x_1, x_2, \dots, x_k\} \cup \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k\}$ as some set of $k' = 2k$ variables. Intuitively, each variable x_i represents a color $i \in [k]$, and each value $j \in [n']$ that can be assigned to x_i can be thought of as the potential choice of v_j^i as the vertex of color i selected into a multicolored clique of size k . We will force the copy \bar{x}_i of each variable x_i to be assigned the value “complementary” to the one assigned to x_i , which will allow us to encode inequalities of the form \leq involving x_i using inequalities of the form \geq involving \bar{x}_i . Moreover, we define $W = k(n' + 1)$. Now, it remains to define the set C .

The set C will consist of two sets of constraints, C^V and C^E (that is, $C = C^V \cup C^E$). Let us first define the set C^V as follows. For all $i \in [k]$ and $j \in [n']$, we have the $\{x_i, \bar{x}_i\}$ -mini-constraint $R_{i,j}^V = \{(x_i, j), (\bar{x}_i, n' - j + 1)\}$. Then, for all $i \in [k]$, we have the constraint $C_i^V = (\{x_i, \bar{x}_i\}, \mathcal{R}_i^V = \{R_{i,j}^V : j \in [n']\})$, whose arity is 2. Next, we define $C^V = \{C_i^V : i \in [k]\}$. Intuitively, this set of constraints, together with the choice of W , will ensure that for all $i \in [k]$, x_i and \bar{x}_i must be assigned complementary values.

Finally, we define the set C^E . We say that two vertices $v_a^i, v_b^j \in V(G)$ have a *conflict* if $i \neq j$ and $\{v_a^i, v_b^j\} \notin E(G)$. For every two conflicting vertices $v_a^i, v_b^j \in V(G)$, we have the constraint $C_{(i,a),(j,b)}^E = (\{x_i, \bar{x}_i, x_j, \bar{x}_j\}, \mathcal{R}_{(i,a),(j,b)}^E = \{(x_i, a + 1), (x_j, b + 1), (\bar{x}_i, n' - a + 2), (\bar{x}_j, n' - b + 2)\})$.² Next, we define $C^E = \{C_{(i,a),(j,b)}^E : v_a^i, v_b^j \in V(G) \text{ have a conflict}\}$. Intuitively, this set of constraints will ensure that a set of vertices selected as implied by some satisfying assignment forms a clique.

Correctness. Let us first prove the forward direction of the correctness of our construction.

LEMMA 3.2. *Let (G, χ, k) be an instance of MULTICOLORED CLIQUE. If (G, χ, k) is a Yes-instance of MULTICOLORED CLIQUE, then $\text{red}(G, \chi, k) = (X, C, n', W)$ is a Yes-instance of 4-MONOTONE MIN-CSP.*

PROOF. Suppose that (G, χ, k) is a Yes-instance of MULTICOLORED CLIQUE, and let C be a multicolored clique in G of size k . For every $i \in [k]$, let $\text{id}(i)$ be the integer in $[n']$ such that $v_{\text{id}(i)}^i \in C$. Then, we define an assignment $\alpha : X \rightarrow [n']$ as follows. For all $i \in [k]$, set $\alpha(x_i) = \text{id}(i)$ and $\alpha(\bar{x}_i) = n' - \text{id}(i) + 1$.

Let us first observe that $\text{cost}(\alpha) = \sum_{i=1}^k (\alpha(x_i) + \alpha(\bar{x}_i)) = k(n' + 1) = W$. Now, note that for all $i \in [k]$, the mini-constraint $R_{i, \text{id}(i)}^V$ is satisfied by α , and, therefore, R_i^V is satisfied by α . Thus, the set C^V is also satisfied by α . Next, consider some constraint $C_{(i,a),(j,b)}^E \in C^E$. Then, we have that the two vertices $v_a^i, v_b^j \in V(G)$ have a conflict, which means that $i \neq j$ and $\{v_a^i, v_b^j\} \notin E(G)$. Since C is a multicolored clique in G of size k , we have that at least one vertex in $\{v_a^i, v_b^j\}$ does not belong

²In the definition of $\mathcal{R}_{(i,a),(j,b)}^E$, if one of the values exceeds n' (e.g., $a + 1 > n'$), simply discard the corresponding mini-constraint from $\mathcal{R}_{(i,a),(j,b)}^E$.

to $V(C)$. Without loss of generality, suppose that this vertex is v_a^i , that is, $\text{id}(i) \neq a$. In this case, either $\text{id}(i) \geq a + 1$, in which case $\alpha(x_i) \geq a + 1$ and then α satisfies $(x_i, a + 1)$, or $\text{id}(i) \leq a - 1$, in which case $\alpha(\bar{x}_i) \geq n' - (a - 1) + 1 = n' - a + 2$ and then α satisfies $(\bar{x}_i, n' - a + 2)$. In both cases, we deduce that α satisfies $C_{(i,a),(j,b)}^E$. Since the choice of this constraint was arbitrary, we have that α satisfies C^E . Overall, we have that α is an assignment of cost at most W that satisfies C , and, therefore, (X, C, n', W) is a Yes-instance of 4-MONOTONE MIN-CSP. \square

We proceed by proving the reverse direction.

LEMMA 3.3. *Let (G, χ, k) be an instance of MULTICOLORED CLIQUE. If $\text{red}(G, \chi, k) = (X, C, n', W)$ is a Yes-instance of 4-MONOTONE MIN-CSP, then (G, χ, k) is a Yes-instance of MULTICOLORED CLIQUE.*

PROOF. Suppose that (X, C, n', W) is a Yes-instance of 4-MONOTONE MIN-CSP, and let α be an assignment of cost at most W that satisfies C . Since α satisfies C^V , we have that for all $i \in [k]$, $\alpha(x_i) + \alpha(\bar{x}_i) \geq n' + 1$. Moreover, since $\text{cost}(\alpha) \leq W$, we have that $\sum_{i=1}^k (\alpha(x_i) + \alpha(\bar{x}_i)) \leq k(n' + 1)$. Thus, we derive that for all $i \in [k]$, $\alpha(x_i) + \alpha(\bar{x}_i) = n' + 1$. For all $i \in [k]$, denote $\text{id}(i) = \alpha(x_i)$, and note that $n' - \text{id}(i) + 1 = \alpha(\bar{x}_i)$. We define C as the graph $G[\{v_{\text{id}(i)}^i : i \in [k]\}]$.

The definition of C directly implies that it is a multicolored graph on k vertices. We now argue that C is also a clique. By way of contradiction, suppose that this claim is false, and therefore, there exist two distinct vertices $v_{\text{id}(i)}^i, v_{\text{id}(j)}^j \in V(C)$ such that $\{v_{\text{id}(i)}^i, v_{\text{id}(j)}^j\} \notin E(G)$. Then, $v_{\text{id}(i)}^i$ and $v_{\text{id}(j)}^j$ have a conflict. Since α satisfies C^E , it in particular satisfies $C_{(i, \text{id}(i)), (j, \text{id}(j))}^E = (\{x_i, \bar{x}_i, x_j, \bar{x}_j\}, \mathcal{R}_{(i, \text{id}(i)), (j, \text{id}(j))}^E)$, where $\mathcal{R}_{(i, \text{id}(i)), (j, \text{id}(j))}^E = \{(x_i, \text{id}(i) + 1)\}, \{(x_j, \text{id}(j) + 1)\}, \{(\bar{x}_i, n' - \text{id}(i) + 2)\}, \{(\bar{x}_j, n' - \text{id}(j) + 2)\}$. In other words, at least one of the following four conditions is satisfied: **(i)** $\alpha(x_i) \geq \text{id}(i) + 1$, which contradicts that $\text{id}(i) = \alpha(x_i)$; **(ii)** $\alpha(x_j) \geq \text{id}(j) + 1$, which contradicts that $\text{id}(j) = \alpha(x_j)$; **(iii)** $\alpha(\bar{x}_i) \geq n' - \text{id}(i) + 2$, which contradicts that $n' - \text{id}(i) + 1 = \alpha(\bar{x}_i)$; **(iv)** $\alpha(\bar{x}_j) \geq n' - \text{id}(j) + 2$, which contradicts that $n' - \text{id}(j) + 1 = \alpha(\bar{x}_j)$. We thus conclude that C is a multicolored clique on k vertices, and, therefore, (G, χ, k) is a Yes-instance of MULTICOLORED CLIQUE. \square

We are now ready to conclude the correctness of Lemma 3.1.

PROOF OF LEMMA 3.1. Suppose, by way of contradiction, that there exists an algorithm \mathcal{A} that solves 4-MONOTONE MIN-CSP in time $f(k) \cdot n^{o(k)}$ for some function f of k . Then, consider the following algorithm \mathcal{B} for MULTICOLORED CLIQUE. Given an instance (G, χ, k) of MULTICOLORED CLIQUE, algorithm \mathcal{B} first constructs the instance $\text{red}(G, \chi, k) = (X, C, n', W)$ of 4-MONOTONE MIN-CSP in polynomial time. Then, it calls algorithm \mathcal{A} with (X, C, n', W) as input and answers the reply given by algorithm \mathcal{A} . By Lemmata 3.2 and 3.3, algorithm \mathcal{B} is correct. Furthermore, as in the output instance, $n' = n/k$ and $k' = 2k$, we have that algorithm \mathcal{B} solves MULTICOLORED CLIQUE in time $f(k) \cdot n^{o(k)}$, which contradicts Proposition 3.1. This concludes the proof. \square

3.2 Reduction to GRAPH COLORING

In this section, we prove Theorem 1 by presenting a reduction from 4-MONOTONE MIN-CSP to GRAPH COLORING.

Construction. Let (X, C, n, W) be an instance of 4-MONOTONE MIN-CSP, where $|X| = 2^k$. Here, we denote $X = \{x_0, x_1, \dots, x_{2^k-1}\}$ (in particular, the first index is 0). We remark that the implicit assumption that $|X|$ is a power of 2 is made without loss of generality, as otherwise we can add some t new dummy variables, where t is the smallest possible integer to ensure that $|X|$ is a power of 2 (which means that at worst, the number of variables is merely doubled). Moreover, without loss of generality, we assume that $W \leq 2^k n$; else it is clear that (X, C, n, W) is a Yes-instance of

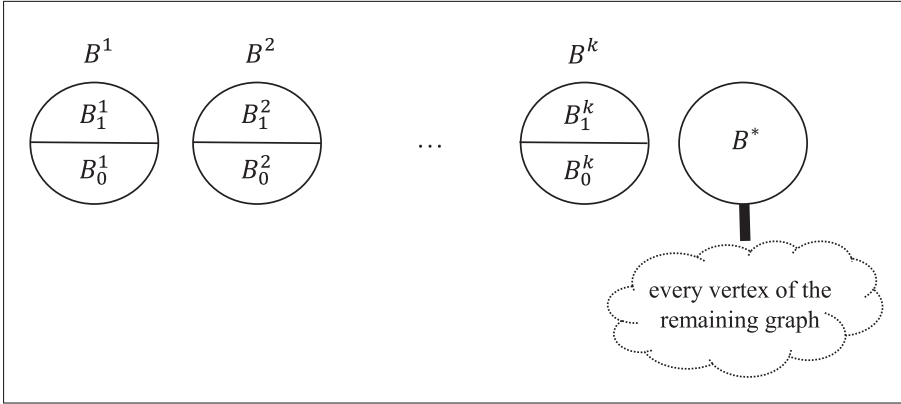


Fig. 1. Assignment Encoder. The thick line is used to denote all edges joining B^* with the remaining vertices of the graph.

4-MONOTONE MIN-CSP (to see this, simply assign n to every variable). Finally, without loss of generality, we assume that every variable $x_i \in X$ belongs to *exactly* one pair in any individual mini-constraint—otherwise, if x_i belongs to more than one pair, then the mini-constraint contains a useless inequality that can be removed, and if x_i belongs to no pair, then we can add the useless pair $(x_i, 0)$. In what follows, we construct an instance $\text{red}(X, C, n, W) = (G, k')$ of GRAPH COLORING, where $k' = 2k + O(1)$ is the neighborhood-width of G . (Note that, as will be formally proved later, the parameter changes from 2^k to $O(k)$).

Assignment Encoder. We first create k vertex-disjoint cliques, B^1, B^2, \dots, B^k , each on $2^k n$ new vertices. We denote $\mathcal{B} = \{B^1, B^2, \dots, B^k\}$. Furthermore, for all $i \in [k]$, we arbitrarily partition B^i into two vertex-disjoint cliques of equal size (that is, $2^{k-1}n$), to which we refer as B_0^i and B_1^i . In addition, we add another clique, called B^* , on $2^k n - W$ new vertices, and denote $\mathcal{B}^* = \mathcal{B} \cup \{B^*\}$. Note that there are no edges between vertices that belong to distinct cliques among the cliques created so far, and we remark that no such edges will be added later. Moreover, whenever we create a new vertex below, we implicitly assume that we also add all edges between that vertex and the vertices in B^* . An illustration of the construction up to this point is given in Figure 1.

Before we proceed with the description of our construction, let us informally explain the intuition behind the definition of these cliques. For every index $i \in [2^k - 1]_0$, let us think of i as the unique ID of the variable x_i . Note that every such ID $i \in [2^k - 1]_0$ can be encoded in binary using only k bits. Intuitively, for all $b \in [k]$, the clique B^b can be thought of as being associated with the b^{st} bit of all IDs, where for specific IDs, B_0^b and B_1^b indicate whether that bit is 0 or 1, respectively. Moreover, for all $i \in [2^k - 1]_0$ and $b \in [k]$, let $\text{bit}(i, b)$ denote the b^{st} bit of the ID i . That is,

$$i = \sum_{b=1}^k \text{bit}(i, b) 2^{b-1}.$$

Accordingly, for all $i \in [2^k - 1]_0$, let us denote the set of cliques that together represent the encoding of i in binary by $\mathcal{B}[i] = \{B_{\text{bit}(i,b)}^b : b \in [k]\}$, and also let us denote the complementary set by $\overline{\mathcal{B}}[i] = \{B_{1-\text{bit}(i,b)}^b : b \in [k]\}$.

We will later ensure (as will be clear in the proof) that all of the cliques in \mathcal{B}^* must be together properly colored using exactly $2^k n$ colors (clearly, they cannot be colored using less than $2^k n$ colors, as every clique $B^b \in \mathcal{B}$ is of the size $2^k n$). The clique B^* can be thought of as a garbage

collector, which forces that at most W colors can be reused to color both vertices in cliques in \mathcal{B} and vertices outside the cliques in \mathcal{B}^* . For the sake of clarity of what follows, we now give a rough (partial) explanation of how the cliques in \mathcal{B} are meant to encode assignments. To this end, let us consider some specific variable $x_i \in X$. Suppose we want to assign some value $v \in [n]_0$ to this variable. Then, the manner to do so is to arbitrarily choose some v vertices in every clique in $\mathcal{B}[i]$, to color the set of the chosen vertices (across all the k cliques in $\mathcal{B}[i]$) using exactly v colors, and to avoid reusing any of these v colors to color any vertex in \mathcal{B}^* . Conversely, to decode the value v assigned to x_i , we compute how many colors have the properties of being used to color a vertex in every clique in $\mathcal{B}[i]$ as well as not being used to color any vertex in \mathcal{B}^* . Importantly, note that the ways in which we encode and decode values of distinct variables are independent of one another—for all distinct $i, j \in [2^k - 1]_0$, a color that appears in all the cliques in $\mathcal{B}[i]$ cannot also appear in all the cliques in $\mathcal{B}[j]$, and vice versa.

Constraint Variable. Let M denote the maximum number of mini-constraints of a constraint in \mathcal{C} . For every constraint $C = (X', \mathcal{R}) \in \mathcal{C}$ and variable $x_i \in X'$, we create a gadget as follows. First, we create a new clique, called $A^{(C,i)}$, on nM new vertices. We arbitrarily partition $A^{(C,i)}$ into $|\mathcal{R}| + 1$ vertex-disjoint cliques, denoted by $A_R^{(C,i)}$ for all $R \in \mathcal{R}$ and $A_\star^{(C,i)}$, where for all $R \in \mathcal{R}$, the clique $A_R^{(C,i)}$ contains n vertices, and the clique $A_\star^{(C,i)}$ contains $n(M - |\mathcal{R}|)$ vertices (this clique might be empty). Now, we add an edge between every vertex in $A^{(C,i)}$ and every vertex that belongs to a clique in $\overline{\mathcal{B}[i]}$. In addition, we create another clique, called $F^{(C,i)}$, on $(n - 1)M$ vertices. We add edges to the graph so that each of the vertices in $F^{(C,i)}$ is adjacent to all vertices in the graph (including those that will be added later) except for the vertices in $A^{(C,i)}$. An illustration of the Constraint Variable gadget is given in Figure 2.

We proceed by presenting a brief intuitive explanation of this gadget. Here, our purpose will be to ensure that $A^{(C,i)}$ can be colored only using colors of the following three types: **(i)** colors of vertices in $F^{(C,i)}$; **(ii)** colors used to decode the value of x_i as explained above; **(iii)** colors of “matching vertices,” which will be defined later. (Observe that due to the existence of edges between the vertices in $F^{(C,i)}$ and any other vertex in the graph excluding those in $A^{(C,i)}$, colors of the first type can, in fact, only be used to color vertices in $F^{(C,i)}$ and $A^{(C,i)}$.) In particular, to be able to properly color $A^{(C,i)}$, there should be at least n colors of the second and third types available to use. Specifically, we will ensure that if we are interested to enforce that $\alpha(x_i) \geq c$ in the context of some assignment α and pair (x_i, c) in a mini-constraint in \mathcal{R} , then exactly $n - c$ colors of the third type will be available, which would mean that at least c colors of the second type should be available.

Mini-Constraint Selector. For every constraint $C = (X', \mathcal{R}) \in \mathcal{C}$, we now present a gadget that aims to encode the selection of a mini-constraint in \mathcal{R} that should be satisfied. For this purpose, we first add one new special vertex, denoted by s^C . Now, for every mini-constraint $R \in \mathcal{R}$, we add an independent set $I^{(C,R)}$ on

$$\sum_{(x_i, c) \in R} (n - c)$$

new vertices that are each adjacent to all the vertices in the cliques in \mathcal{B} (in addition to the vertices in \mathcal{B}^* and cliques of the form $F^{(C,i)}$). Denote $\mathcal{I}^C = \{I^{(C,R)} : R \in \mathcal{R}\}$. We add an edge between s^C and every vertex in the graph (including those that will be added later) except for the vertices in the independent sets in \mathcal{I}^C . Moreover, for all distinct $R, R' \in \mathcal{R}$, we add an edge between every vertex in $I^{(C,R)}$ and every vertex in $I^{(C,R')}$. For every $R \in \mathcal{R}$, let us now turn to refine the independent set $I^{(C,R)}$ by considering subsets of it. For every $i \in [2^k - 1]_0$ such that $R \in \mathcal{R}$, let $I_i^{(C,R)}$ denote a

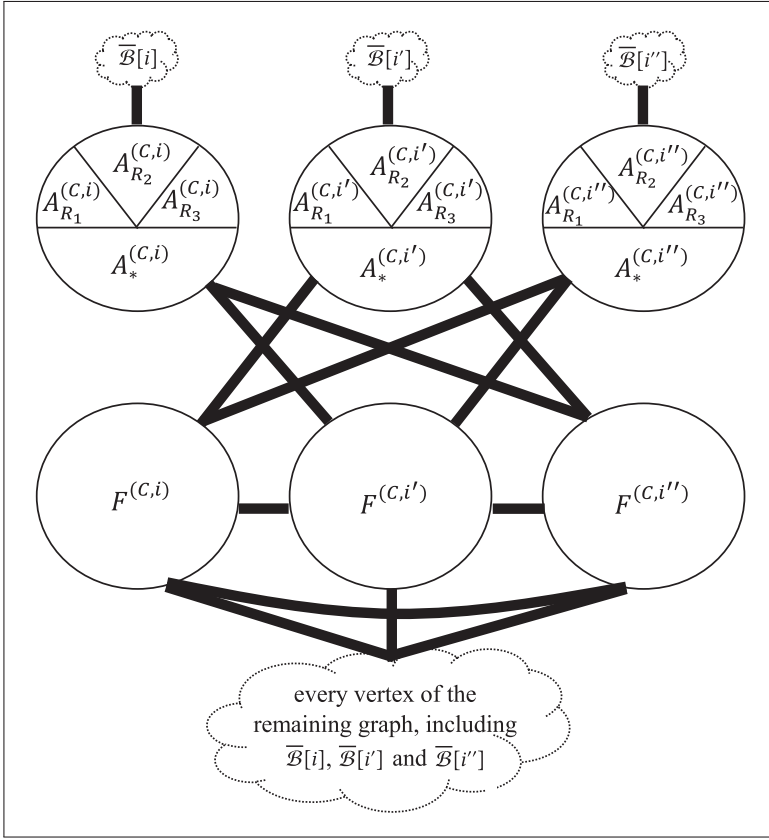


Fig. 2. Constraint Variable; $C = (X', \mathcal{R}), X' = \{i, i', i''\}, \mathcal{R} = \{R_1, R_2, R_3\}$. The thick lines are used to denote all edges joining the cliques $A^{(C,i)}, A^{(C,i')}, A^{(C,i'')}, F^{(C,i)}, F^{(C,i')},$ and $F^{(C,i'')}$ with each other and the remaining vertices of the graph.

subset of $I^{(C,R)}$ of size $(n - c)$ where c is the unique integer in $[n]_0$ satisfying $(x_i, c) \in R$, so that for all distinct $i, j \in [2^k - 1]_0$, it holds that $I_i^{(C,R)} \cap I_j^{(C,R)} = \emptyset$. Clearly, as

$$|I^{(C,R)}| = \sum_{(x_i, c) \in R} (n - c),$$

we have that every vertex in $I^{(C,R)}$ belongs to exactly one independent set $I_i^{(C,R)}$. An illustration of the Mini-Constraint Selector gadget is given in Figure 3.

Let us now explain the intuition underlying the construction of this gadget. To this end, first observe that since s^C is adjacent to all the vertices in the graph apart from those in the independent sets in \mathcal{I}^C , it would definitely have a “new” color. As vertices in distinct independent sets in \mathcal{I}^C are adjacent, only one independent set can have vertices colored with the same color as s^C . Moreover, as our color set is the resource we aim to use as little as possible, it would be possible to assume that exactly one independent set has vertices colored with the same color as s^C and, furthermore, all the vertices of this independent set have the same color. The mini-constraint $R \in \mathcal{R}$ such that $I^{(C,R)}$ is the independent set that “won” this unique color is the one to be thought of as the mini-constraint in \mathcal{R} that we should satisfy. Roughly speaking, we note that $I^{(C,R)}$ is thought of as one unit, in

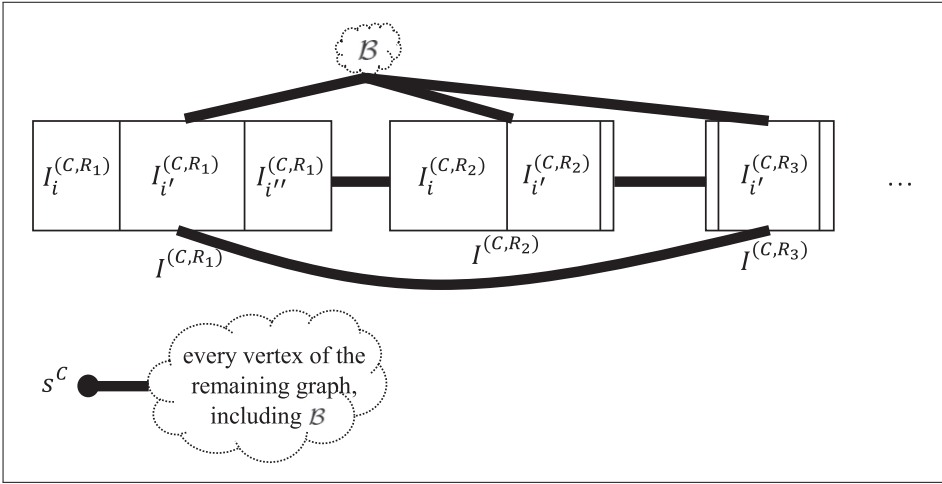


Fig. 3. Mini-Constraint Selector. The thick lines are used to denote all edges joining s^C and the independent sets $I_i^{(C,R_1)}$, $I_{i'}^{(C,R_2)}$, $I_{i'}^{(C,R_3)}$, \dots with each other and the remaining vertices of the graph.

the sense that all the variables that occur in R will be affected *simultaneously* by the selection of R (using the Matching Vertices gadget defined below), which is done to comply with the demand that if a mini-constraint is to be satisfied, all of the inequalities corresponding to its pairs must be satisfied simultaneously.

Matching Edges and Vertices. For every constraint $C = (X', \mathcal{R}) \in \mathcal{C}$, we now add a gadget that relates the Constraint Variable gadgets associated with C to the Mini-Constraint Selector gadgets associated with C . For this purpose, for every (existing) clique of the form $A_R^{(C,i)}$ for some $i \in [2^k - 1]_0$ and $R \in \mathcal{R}$, we perform the following operations. We first let $\tilde{A}_R^{(C,i)}$ denote some arbitrarily chosen subclique of $A_R^{(C,i)}$ on $(n - c)$ vertices where $(n - c) = |I_i^{(C,R)}|$. Now, we add a set of $(n - c)$ new edges to G , denoted by $M_R^{(C,i)}$, that together form an arbitrarily chosen perfect matching of size $(n - c)$ in $G[I_i^{(C,R)} \cup V(\tilde{A}_R^{(C,i)})]$, where each new edge has one endpoint in $I_i^{(C,R)}$ and the other endpoint in $\tilde{A}_R^{(C,i)}$. Finally, we add $(n - c)$ new vertices, denoted by v^e for all $e \in M_R^{(C,i)}$, and add edges between each v^e and all the vertices in G apart from the two vertices that are the endpoints of the edge e . Let us denote $\widehat{M}_R^{(C,i)} = \{v^e : e \in M_R^{(C,i)}\}$. An illustration of this construction is given in Figure 4.

Intuitively, the addition of the new sets $M_R^{(C,i)}$ and $\widehat{M}_R^{(C,i)}$ aims to relate $A_R^{(C,i)}$ and $I_i^{(C,R)}$ as follows. First, observe that the color of each of the new vertices $v^e \in \widehat{M}_R^{(C,i)}$ can be reused only to color one of the endpoints of e . In case $I_i^{(C,R)}$ is colored with the same color as s^C —that is, R is the mini-constraint in \mathcal{R} that we would like to satisfy—we are “free” to reuse the colors of the vertices in $\widehat{M}_R^{(C,i)}$ in order to color the vertices in $\tilde{A}_R^{(C,i)}$, and, otherwise, we are “forced” to spend these colors on the vertices in $I_i^{(C,R)}$. Roughly speaking, notice that the larger $c = n - |I_i^{(C,R)}|$ is, the harder it is for an assignment α to ensure that $\alpha(x_i) \geq c$ as required to satisfy $(x_i, c) \in R$; and, indeed, the larger c is, the smaller the set of “free” colors is since its size is $|\widehat{M}_R^{(C,i)}| = n - c$. Recalling the three types of colors that can be used to color $A^{(C,i)}$ (in the description of the Constraint Variable gadgets), and combining this with our last note, it would be possible to formally argue in our proofs that if we choose to satisfy the mini-constraint R while overall using only “few” colors, the $(n - c)$

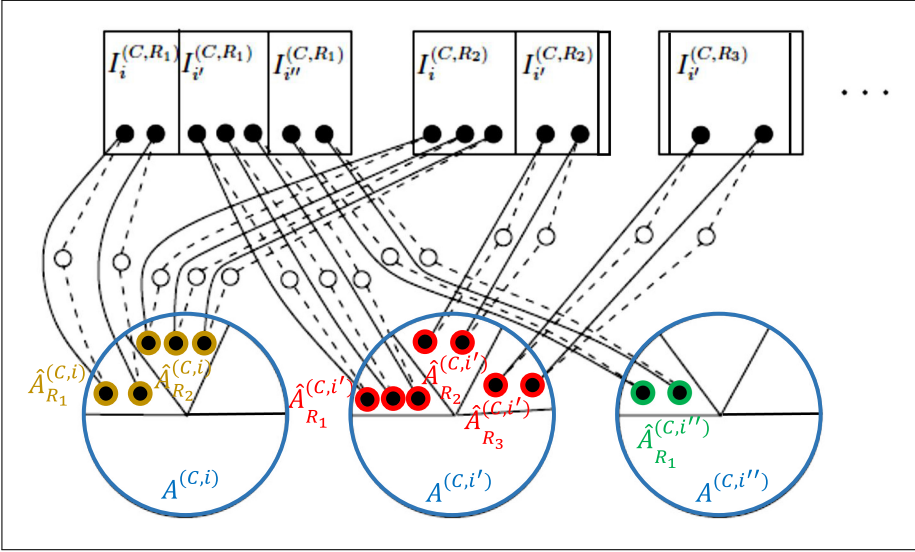


Fig. 4. Matching Edges and Vertices. The white bullets are used to depict the vertices v_e and the incident dashed lines show non-edges.

free colors of $\widehat{M}_R^{(C,i)}$ would have to be complemented with c colors of type (ii) in order to properly color $A^{(C,i)}$. As desired, this means that we would be able to argue that if $I^{(C,R)}$ is the independent set reusing the color of s^C , then for all $(x_i, c) \in R$, the assignment α decoded from the coloring will satisfy $\alpha(x_i) \geq c$.

Chromatic Number of a Yes-instance. Denote

$$\eta = 2^k n + \left((n-1)M \sum_{(X', \mathcal{R}) \in \mathcal{C}} |X'| \right) + |C| + \left(\sum_{(X', \mathcal{R}) \in \mathcal{C}} \sum_{R \in \mathcal{R}} \sum_{(x_i, c) \in R} (n-c) \right).$$

(This value is polynomial in the input size because $|X| = 2^k$.) Informally, this value would be the threshold for the chromatic number of the output graph according to which we will determine whether the input instance of 4-MONOTONE MIN-CSP is a Yes-instance or a No-instance.

Correctness. Let us first prove the forward direction of the correctness of our construction.

LEMMA 3.4. *Let (X, C, n, W) be an instance of 4-MONOTONE MIN-CSP with $|X| = 2^k$. If (X, C, n, W) is a Yes-instance of 4-MONOTONE MIN-CSP, then the chromatic number of G in $\text{red}(X, C, n, W) = (G, k')$ is at most η .*

PROOF. Suppose that (X, C, n, W) is a Yes-instance of 4-MONOTONE MIN-CSP, and let α be an assignment of cost at most W that satisfies C . Without loss of generality, we can assume that $\text{cost}(\alpha) = W$; we otherwise can increase the value assigned to some variables so that this condition will be satisfied. In what follows, we construct a proper coloring $\chi : V(G) \rightarrow [\eta]$. This would imply that the chromatic number of G is at most η , which would conclude the proof. Recall that $\mu = \eta - 2^k n$. First, we use $\mu = ((n-1)M \cdot \sum_{(X', \mathcal{R}) \in \mathcal{C}} |X'|) + |C| + (\sum_{(X', \mathcal{R}) \in \mathcal{C}} \sum_{R \in \mathcal{R}} \sum_{(x_i, c) \in R} (n-c))$ colors, say the colors in $[\mu]$, to (arbitrarily) color all the vertices in the set $(\bigcup_{C=(X', \mathcal{R}) \in \mathcal{C}} \bigcup_{x_i \in X'} V(F^{(C,i)}) \cup \{s^C : C \in \mathcal{C}\} \cup (\bigcup_{C=(X', \mathcal{R}) \in \mathcal{C}} \bigcup_{x_i \in X'} \widehat{M}_R^{(C,i)}))$, whose size

is exactly μ , with distinct colors. Since we have not reused any color so far, it is clear that we have also not colored the endpoints of any edge with the same color so far.

We proceed by using W new colors, that is, the colors in $[\mu + W] \setminus [\mu] \subseteq [\eta]$ (note that $2^k n \geq W$ and, hence, $\mu + W \leq \eta$), to color some of the vertices in the cliques in \mathcal{B} . For every index $i \in [2^k - 1]_0$ and $B \in \mathcal{B}[i]$, let $\text{alloc}(i, B)$ be some (arbitrarily chosen) set of $\alpha(x_i)$ vertices in B , so that for all distinct $i, j \in [k]$ and $B \in \mathcal{B}[i] \cap \mathcal{B}[j]$, $\text{alloc}(i, B) \cap \text{alloc}(j, B) = \emptyset$. Since for all $i \in [k]$, the size of each clique $B \in \mathcal{B}[i]$ is $2^{k-1}n$ (recall that such B is only a “half” of a clique in \mathcal{B}) and there exist at most 2^{k-1} indices $j \in [2^k - 1]_0$ in total such that $B \in \mathcal{B}[j]$, as well as since the maximum value assigned by α is n , we have that there is a sufficient number of vertices to ensure that alloc can be well-defined. Moreover, for all $i \in [2^k - 1]_0$, let $\text{col}(i)$ denote some (arbitrarily chosen) set of $\alpha(x_i)$ colors in $[\mu + W] \setminus [\mu]$, so that for all distinct $i, j \in [k]$, $\text{col}(i) \cap \text{col}(j) = \emptyset$. Since $\text{cost}(\alpha) \leq W$, there is a sufficient number of colors in $[\mu + W] \setminus [\mu]$ to ensure that col can be well-defined. Now, for all $i \in [2^k - 1]_0$ and $B \in \mathcal{B}[i]$, we (arbitrarily) color all the vertices in $\text{alloc}(i, B)$ with distinct colors from $\text{col}(i)$. Clearly, all the vertices of the same clique in \mathcal{B} that we have colored so far received distinct colors (because for all distinct $i, j \in [k]$, $\text{col}(i) \cap \text{col}(j) = \emptyset$), and there are no edges between vertices in different cliques in \mathcal{B} . Therefore, it still holds that we have not colored the endpoints of any edge with the same color so far.

Note that we have not yet used any of the colors in $[\eta] \setminus [\mu + W]$, and that for every color in $[\mu + W] \setminus [\mu]$, each clique in \mathcal{B} has exactly one vertex with that color (because $\text{cost}(\alpha) = W$). Since the size of each clique in \mathcal{B} is exactly $2^k n$ and $\eta = \mu + 2^k n$, for every clique in \mathcal{B} , individually, we can use the remaining colors in $[\eta] \setminus [\mu + W]$ to color every yet uncolored vertex with a distinct color. Moreover, since $|V(B^*)| = 2^k n - W$ and there are no edges between vertices in B^* and vertices in the cliques in \mathcal{B} , we can also color every vertex in B^* with a distinct color from $[\eta] \setminus [\mu + W]$ so that still no edge has both endpoints colored with the same color.

In what follows, we proceed to color the vertices in all the cliques of the form $A^{(C,i)}$ as well as in all the independent sets of the form $I^{(C,R)}$. Here, we will only consider colors already used—in particular, when we color a vertex v , we will say that v is to be colored with the color of some previously colored vertex. Thus, it will be clear that, overall, we do not exceed our budget of colors η . The point that we will have to argue about each time is that each newly colored vertex is not adjacent to a vertex with the same color. To this end, we consider the constraints $C = (X', \mathcal{R}) \in \mathcal{C}$ one by one, and in each iteration color the vertices in $A^{(C,i)}$ for all $x_i \in X'$ as well as I^C . Since for distinct constraints $C = (X', \mathcal{R}), \widehat{C} = (\widehat{X}, \widehat{\mathcal{R}}) \in \mathcal{C}$, there is no edge between a vertex in $A^{(C,i)}$ for some $x_i \in X'$ or in I^C and a vertex in $A^{(\widehat{C},j)}$ for some $x_j \in \widehat{X}$ or in $I^{\widehat{C}}$, we can indeed analyze each constraint separately. Next, we fix some constraint $C = (X', \mathcal{R}) \in \mathcal{C}$. Moreover, we let R be a mini-constraint in \mathcal{R} that is satisfied by α , whose existence follows from the fact that α satisfies C . For all $i \in [2^k - 1]_0$ such that $x_i \in X'$, let c_i denote the (unique) integer in $[n]_0$ satisfying $(x_i, c_i) \in R$.

Let us begin by coloring the vertices in the independent sets in I^C . To this end, we first color all the vertices in $I^{(C,R)}$ with the color of s^C . Since the color of s^C is not used by any other vertex and $I^{(C,R)} \cup \{s^C\}$ is an independent set, no edge has both endpoints colored with the same color. For every $R' \in \mathcal{R} \setminus \{R\}$ and for every edge $e \in M^{(C,R')}$, we color the endpoint of e in $I^{(C,R')}$ with the color of v^e (which is adjacent to neither endpoint of e and whose color was not reused before). Clearly, we have thus colored all the vertices in all the independent sets in $I^C \setminus \{I^{(C,R)}\}$ so that no edge has both endpoints colored with the same color.

It remains to color the vertices in $A^{(C,i)}$. First, for every edge $e \in M^{(C,R)}$, we color the endpoint of e in $\widehat{A}_R^{(C,i)}$ with the color of v^e . In this context, note that v^e is adjacent to neither endpoint of e , and its color was not reused before, as the endpoint of e in $I^{(C,R)}$ was colored by s^C . So far, we have colored $|M^{(C,R)}| = n - c_i$ vertices among the nM vertices in $A^{(C,i)}$. Since α satisfies R , we have

that $\alpha(x_i) \geq c_i$, and, therefore, $|\text{col}(i)| \geq c_i$. Note that all the vertices colored by colors in $\text{col}(i)$ belong to cliques in $\mathcal{B}[i]$, and that no vertex in any of these cliques is adjacent to any vertex in $A^{(C,i)}$ (only the vertices in the cliques in $\overline{\mathcal{B}}[i]$ are adjacent to the vertices in $A^{(C,i)}$). Therefore, we can safely color c_i additional vertices in $A^{(C,i)}$ using the colors in $\text{col}(i)$. The remaining $(n-1)M$ vertices in $A^{(C,i)}$ are now colored using the $(n-1)M$ colors used to color the vertices in $F^{(C,i)}$. Thus, we have overall ensured that no edge has both endpoints colored with the same color. This completes the proof. \square

Toward the proof of the reverse direction, we first establish several definitions and claims. We begin by defining an assignment decoded from a proper coloring of a graph outputted by our reduction.

Definition 3.1. Let (X, C, n, W) be an instance of 4-MONOTONE MIN-CSP with $|X| = 2^k$, and denote $\text{red}(X, C, n, W) = (G, k')$. Let χ be a proper coloring of G . Then, the assignment $\alpha_\chi : X \rightarrow [n]_0$ is defined as follows. For all $i \in [2^k - 1]_0$, denote $\text{col}_\chi(i) = \{j \in \text{ima}(\chi) : \text{every clique in } \mathcal{B}[i] \text{ has a vertex colored } j \text{ by } \chi\} \setminus \chi(V(B^*))$. Then, for all $x_i \in X$, $\alpha(x_i) \triangleq |\text{col}_\chi(i)|$.

It will also be convenient for us to use the following notation: In the context of an output $\text{red}(X, C, n, W) = (G, k')$, we denote $D = (\bigcup_{C=(X', \mathcal{R}) \in \mathcal{C}} \bigcup_{x_i \in X'} V(F^{(C,i)}) \cup \{s^C : C \in \mathcal{C}\} \cup (\bigcup_{C=(X', \mathcal{R}) \in \mathcal{C}} \bigcup_{x_i \in X'} \widehat{M}_R^{(C,i)})$.

OBSERVATION 3.1. *Let (X, C, n, W) be an instance of 4-MONOTONE MIN-CSP with $|X| = 2^k$, and denote $\text{red}(X, C, n, W) = (G, k')$. Then, for all $i \in [k]$, any two distinct vertices in $V(B^i) \cup D$ are assigned distinct colors by χ .*

PROOF. For all $i \in [k]$, the subgraph of G induced by $V(B^i) \cup D$ forms a clique, and, therefore, the observation is correct. \square

By using Observation 3.1, we derive the following result.

LEMMA 3.5. *Let (X, C, n, W) be an instance of 4-MONOTONE MIN-CSP with $|X| = 2^k$, and denote $\text{red}(X, C, n, W) = (G, k')$. Let χ be a proper coloring of G such that $\text{ima}(\chi) \subseteq [\eta]$. For all $i \in [k]$, $\chi(V(B^i)) = [\eta] \setminus \chi(D)$ and $\chi(V(B^*)) \subseteq \chi(V(B^i))$.*

PROOF. First, notice that $|D| = \eta - 2^k n$, and that for all $i \in [k]$, $|B^i| = 2^k n$. By Observation 3.1, for all $i \in [k]$, we have that $\chi(B^i) \subseteq \text{ima}(\chi) \setminus \chi(D)$. However, since $\text{ima}(\chi) \subseteq [\eta]$, this implies that for all $i \in [k]$, indeed $\chi(B^i) = \text{ima}(\chi) \setminus \chi(D)$. Since every vertex in B^* is adjacent to all vertices in G apart from those in the cliques in \mathcal{B} , we have that for any $i \in [k]$, indeed also $\chi(V(B^*)) \subseteq \chi(V(B^i))$. \square

At this point, we are already able to analyze the cost of a decoded assignment.

LEMMA 3.6. *Let (X, C, n, W) be an instance of 4-MONOTONE MIN-CSP with $|X| = 2^k$, and denote $\text{red}(X, C, n, W) = (G, k')$. Let χ be a proper coloring of G such that $\text{ima}(\chi) \subseteq [\eta]$. Then, $\text{cost}(\alpha_\chi) \leq W$.*

PROOF. By the definition of α_χ (Definition 3.1), it holds that $\text{cost}(\alpha_\chi) = \sum_{i=0}^{2^k-1} |\text{col}_\chi(i)|$. Thus, to prove that the lemma is correct, we need to show that $\sum_{i=0}^{2^k-1} |\text{col}_\chi(i)| \leq W$. For this purpose, first note that for all distinct $i, j \in [2^k - 1]_0$, it holds that $\mathcal{B}[i] \neq \mathcal{B}[j]$, and, hence, from the definition of $\text{col}(\cdot)$, we have that $\text{col}(i) \cap \text{col}(j) = \emptyset$. Thus, $\sum_{i=0}^{2^k-1} |\text{col}_\chi(i)| = |\bigcup_{i=0}^{2^k-1} \text{col}(i)|$. Now, note that for any $i \in [k]$, $\bigcup_{i=0}^{2^k-1} \text{col}(i) \subseteq \chi(V(B^i)) \setminus \chi(V(B^*))$, $|V(B^i)| = 2^k n$ and $|V(B^*)| = 2^k n - W$. Moreover, by Lemma 3.5, for any $i \in [k]$, $\chi(V(B^*)) \subseteq \chi(V(B^i))$. Therefore, for any $i \in [k]$, $|\bigcup_{i'=0}^{2^k-1} \text{col}(i')| = |V(B^i)| - |V(B^*)| = W$. We thus have that $\text{cost}(\alpha_\chi) \leq W$. \square

We proceed by defining a special kind of proper coloring with respect to the graphs outputted by our reduction.

Definition 3.2. Let (X, C, n, W) be an instance of 4-MONOTONE MIN-CSP with $|X| = 2^k$, and denote $\text{red}(X, C, n, W) = (G, k')$. A function χ is a *nice coloring* of G if it is a proper coloring of G , $\text{ima}(\chi) \subseteq [\eta]$, and for all $C = (X', \mathcal{R}) \in C$, the two following conditions hold.

- (1) There exists exactly one mini-constraint in \mathcal{R} , denote by R_χ^C , such that all the vertices in $I^{(C, R_\chi^C)}$ have the same color as s^C .
- (2) For all $R \in \mathcal{R} \setminus \{R_\chi^C\}$ and $v \in I^{(C, R)}$, the color of v is the same as v^e where e is the (unique) edge in $\bigcup_{x_i \in X'} M_R^{(C, i)}$ incident to v .

LEMMA 3.7. Let (X, C, n, W) be an instance of 4-MONOTONE MIN-CSP with $|X| = 2^k$, and denote $\text{red}(X, C, n, W) = (G, k')$. If there exists a proper coloring χ of G such that $\text{ima}(\chi) \subseteq [\eta]$, then there also exists a nice coloring $\widehat{\chi}$ of G .

PROOF. Let χ be a proper coloring of G such that $\text{ima}(\chi) \subseteq [\eta]$. Consider some constraint $C = (X', \mathcal{R}) \in C$. First, recall that for every two distinct $R, R' \in \mathcal{R}$, all the vertices in $I^{(C, R)}$ are adjacent to all the vertices in $I^{(C, R')}$. Thus, there exists at most one $R \in \mathcal{R}$ such that at least one vertex in $I^{(C, R)}$ has the same color as s^C . Let R_χ^C denote the mini-constraint with this property, where if no such mini-constraint exists, arbitrarily choose some mini-constraint from \mathcal{R} . Now, since s^C is adjacent to all the vertices that do not belong to $I^{(C, R)}$ for some $R \in \mathcal{R}$, we can recolor all the vertices in $I^{(C, R_\chi^C)}$ with the color of s^C , so that the resulting coloring $\widehat{\chi}$ remains a proper coloring, and, clearly, it still holds that only colors from $[\eta]$ are used. To complete the proof, it remains to show that $\widehat{\chi}$ satisfies the second property in the list. For this purpose, consider some $R \in \mathcal{R} \setminus \{R_\chi^C\}$ and $v \in I^{(C, R)}$. Observe that v is adjacent to all the vertices in $D \cup (\bigcup_{i \in [k]} V(B^i))$ apart from s^C and v^e where e is the (unique) edge in $\bigcup_{x_i \in X'} M_R^{(C, i)}$ incident to v . By Lemma 3.5 and since $\text{ima}(\widehat{\chi}) \subseteq [\eta]$, we get that v has the same color as either s^C or v^e . However, as we have already argued that no vertex in $I^{(C, R)}$ has the same color as s^C , we conclude that v necessarily has the same color as v^e . As the choice of C was arbitrary, the above modification can be done for every constraint in C individually. This completes the proof. \square

Now, we present two claims that shed light on the usefulness of analyzing nice colorings. To this end, it will be convenient to use the following notation: In the context of an output $\text{red}(X, C, n, W) = (G, k')$ and a nice coloring χ of G , for all $C = (X', \mathcal{R}) \in C$ and $x_i \in X'$, let $c_\chi^{(C, i)}$ denote a unique integer in $[n]_0$ satisfying $(x_i, c_\chi^{(C, i)}) \in R_\chi^C$.

LEMMA 3.8. Let (X, C, n, W) be an instance of 4-MONOTONE MIN-CSP with $|X| = 2^k$, and denote $\text{red}(X, C, n, W) = (G, k')$. Let χ be a nice coloring of G . For all $\widehat{C} = (\widehat{X}, \widehat{\mathcal{R}}) \in C$ and $x_{\widehat{t}} \in \widehat{X}$, at most $n - c_\chi^{(\widehat{C}, \widehat{i})}$ vertices in $A^{(\widehat{C}, \widehat{i})}$ are colored with a color that is also used for vertices in $\bigcup_{C=(X', \mathcal{R}) \in C} \bigcup_{x_i \in X'} \widehat{M}_R^{(C, i)}$.

PROOF. Let us fix some $\widehat{C} = (\widehat{X}, \widehat{\mathcal{R}}) \in C$ and $x_{\widehat{t}} \in \widehat{X}$. First, note that all the vertices in $A^{(\widehat{C}, \widehat{i})}$ are adjacent to all the vertices in $\bigcup_{C=(X', \mathcal{R}) \in C \setminus \{\widehat{C}\}} \bigcup_{x_i \in X'} \widehat{M}_R^{(C, i)}$, and, therefore, they can clearly not be colored as the vertices in this set. Moreover, for all $R \in \widehat{\mathcal{R}} \setminus \{R_\chi^{\widehat{C}}\}$ and $v \in I^{(\widehat{C}, R)}$, we have that v is colored as v^e where e is the edge in $\bigcup_{x_i \in X'} M_R^{(C, i)}$ incident to v (as χ is a nice coloring of G), and we note that every vertex in $A^{(\widehat{C}, \widehat{i})}$ is incident to either v or v^e . Thus, among the vertices

in $\bigcup_{C=(X',\mathcal{R})\in\mathcal{C}} \bigcup_{x_i\in X'} \widehat{M}_R^{(C,i)}$, the vertices in $A^{(\widehat{C},\widehat{i})}$ can only be colored as the vertices in $\widehat{M}_{R\widehat{C}}^{(\widehat{C},\widehat{i})}$. Since $|\widehat{M}_{R\widehat{C}}^{(\widehat{C},\widehat{i})}| = n - c_\chi^{(\widehat{C},\widehat{i})}$, this completes the proof. \square

LEMMA 3.9. *Let (X, C, n, W) be an instance of 4-MONOTONE MIN-CSP with $|X| = 2^k$, and denote $\text{red}(X, C, n, W) = (G, k')$. Let χ be a nice coloring of G . For all $C = (X', \mathcal{R}) \in \mathcal{C}$ and $x_i \in X'$, at least $c_\chi^{(C,i)}$ vertices in $A^{(C,i)}$ are colored using colors $j \in [\eta]$ with the following property: Every clique in $\mathcal{B}[i]$ has a vertex colored j , and no vertex in B^* is colored j .*

PROOF. Fix some $C = (X', \mathcal{R}) \in \mathcal{C}$ and $x_i \in X'$. By Observation 3.1 and since $\text{ima}(\chi) \subseteq [\eta]$, all the vertices in $A^{(C,i)}$ must be colored as vertices in $B^{\widehat{i}} \cup D$ for any $\widehat{i} \in [k]$. However, by Lemma 3.8 and since every vertex in $A^{(C,i)}$ is adjacent to every vertex in the cliques $F^{(C',i')}$ for $(C', i') \neq (C, i)$ and $B \in \overline{\mathcal{B}}[i]$, we have that all the vertices in $A^{(C,i)}$, apart from at most $n - c_\chi^{(C,i)}$ vertices, are colored as either some vertex in the clique $F^{(C,i)}$ or with a color that is present in every clique in $\mathcal{B}[i]$. Because $|F^{(C,i)}| = (n-1)M$ and $|A^{(C,i)}| = nM$, we further derive that at least $c_\chi^{(C,i)}$ vertices in $A^{(C,i)}$ are colored with some color that is present in every clique in $\mathcal{B}[i]$. Since every vertex in $A^{(C,i)}$ is adjacent to every vertex in B^* , the correctness of the lemma follows. \square

Finally, we are ready to prove the reverse direction.

LEMMA 3.10. *Let (X, C, n, W) be an instance of 4-MONOTONE MIN-CSP with $|X| = 2^k$. If the chromatic number of G in $\text{red}(X, C, n, W) = (G, k')$ is at most η , then (X, C, n, W) is a Yes-instance of 4-MONOTONE MIN-CSP.*

PROOF. Suppose that the chromatic number of G is at most η . By Lemma 3.7, G has a nice coloring χ . Let α denote the assignment $\alpha_\chi : X \rightarrow [n]_0$. By Lemma 3.6, we have that $\text{cost}(\alpha) \leq W$. Now, let us consider some $C = (X', \mathcal{R}) \in \mathcal{C}$. Then, Lemma 3.9 directly implies that for all $x_i \in X'$, $|\text{col}(i)| \geq c_\chi^{(C,i)}$. However, this means that $\alpha(x_i) \geq c_\chi^{(C,i)}$, and, therefore, α satisfies R_χ^C . In turn, this means that α satisfies C . Since the choice of C was arbitrary, we have that α satisfies C . Hence, (X, C, n, W) is a Yes-instance of 4-MONOTONE MIN-CSP. \square

3.3 Clique-width

Toward the proof of Theorem 1, it remains to bound the clique-width of the output graph.

LEMMA 3.11. *Let (X, C, n, W) be an instance of 4-MONOTONE MIN-CSP with $|X| = 2^k$. The neighborhood-width of G in $\text{red}(X, C, n, W) = (G, k')$ is at most $2k + \mathcal{O}(1)$.*

PROOF. We define an ordering σ on $V(G) = \{v_1, v_2, \dots, v_{n'}\}$ as follows. We let the first $2^k n(k+1) - W$ vertices in this order be all the vertices in the cliques in \mathcal{B}^* , where the internal order among them is arbitrary. Notice that B^* as well as every clique of the form B_b^i is a module in G . Since there are exactly $2k+1$ such cliques, we have that for all $i \in [2^k n(k+1) - W]$, $|EQ(G, \sigma, i)| \leq 2k+1$. Let us denote the set of vertices inserted so far by D^0 .

Let us denote $C = \{C_1, C_2, \dots, C_m\}$, and for all $j \in [m]$, denote $C_j = (X_j, \mathcal{R}_j)$. For $j = 1, 2, \dots, m$, we will consecutively insert all the vertices in $D^j \triangleq (\bigcup_{x_i \in X_j} V(A^{(C_j,i)}) \cup V(F^{(C_j,i)})) \cup \{s^{C_j}\} \cup (\bigcup_{R \in \mathcal{R}_j} I^{(C_j,R)}) \cup (\bigcup_{x_i \in X_j} \bigcup_{R \in \mathcal{R}_j} \widehat{M}_R^{(C_j,i)})$ in an order defined as follows. Fix some $j \in [m]$, and let t be the number of vertices inserted so far, that is, the vertices in $D^{j'}$ for all $0 \leq j' < j$. Now, note that $\bigcup_{j'=1}^{j-1} D^{j'}$ consists of two modules with respect to $V(G) \setminus (\bigcup_{j'=0}^{j-1} D^{j'})$ —namely, $\bigcup_{j' < j} ((\bigcup_{x_i \in X_{j'}} V(A^{(C_{j'},i)}) \cup (\bigcup_{R \in \mathcal{R}_{j'}} I^{(C_{j'},R)}))$ and

$\bigcup_{j' < j} (V(F^{(C_{j',i})})) \cup \{s^{C_j}\} \cup (\bigcup_{x_i \in X_j} \bigcup_{R \in \mathcal{R}_j} \widehat{M}_R^{(C_{j',i})})$. Therefore, $|EQ(G, \sigma, t)| \leq 2k + 3$. We now insert s^{C_j} . Thus, $|EQ(G, \sigma, t + 1)| \leq 2k + 4$. Next, for all $x_i \in X_j$, we insert all the vertices of the clique $G[V(A^{(C_{j,i})}) \setminus (\bigcup_{R \in \mathcal{R}_j} \widehat{A}_R^{(C_{j,i})})]$ in an arbitrary order and then all the vertices of the clique $F^{(C_{j,i})}$ in an arbitrary order (where vertices of the same clique appear consecutively). Since the arity of \mathcal{R}_j is at most 4, we have thus inserted at most eight cliques. Moreover, observe that each one of these cliques is a module with respect to $V(G) \setminus D^0$. Let t' denote the total number of vertices of these cliques. Then, we so far have that for all $i' \in [t + t' + 1]$, $|EQ(G, \sigma, i')| \leq 2k + 12$.

Let us denote $\mathcal{R}_j = \{R_1, R_2, \dots, R_r\}$. For $p = 1, 2, \dots, r$ (outer loop) and $q = 1, 2, \dots, |X_j|$ (inner loop), we will consecutively insert all the vertices in $D_{p,q}^j \triangleq V(\widehat{A}_{R_p}^{(C_j,q)}) \cup I_q^{(C_j,R_p)} \cup \widehat{M}_{R_p}^{(C_j,q)}$ in an order defined as follows. Fix some $p \in [r]$ and $q \in [|X_j|]$, and let \widehat{t} be the total number of vertices inserted so far, that is, the vertices in $D^{j'}$ for all $0 \leq j' < j$ as well as the vertices in $D_{p',q'}^j$ where $(p', q') < (p, q)$; that is, either $1 \leq p' < p$ or both $p' = p$ and $1 \leq q' < q$. Now, note that $\bigcup_{(p',q') < (p,q)} D_{p',q'}^j$ consists of seven modules with respect to $V(G) \setminus (\bigcup_{j'=0}^{j-1} D^{j'}) \cup (\bigcup_{(p',q') < (p,q)} D_{p',q'}^j)$ —namely, $\bigcup_{p'} V(\widehat{A}_{R_p}^{(C_j,q')})$ for each $q' \leq |X_j|$ where p' ranges over all values in $[p]$ such that $(p', q') < (p, q)$ (four modules since $|X_j| \leq 4$), $\bigcup_{p' < p} \bigcup_{q' \in [|X_j|]} I_{q'}^{(C_j,R_{p'})}$ and $\bigcup_{q' < q} I_{q'}^{(C_j,R_p)}$ (two modules), and $\bigcup_{(p',q') < (p,q)} \widehat{M}_{R_p}^{(C_j,q')}$ (one module). Therefore, $|EQ(G, \sigma, \widehat{t})| \leq 2k + 19$.

Finally, let us denote $M_{R_p}^{(C_j,q)} = \{e_1, e_2, \dots, e_s\}$. For $\ell = 1, 2, \dots, s$, we consecutively insert the vertex $v^{e_\ell} \in \widehat{M}_{R_p}^{(C_j,q)}$ and the two endpoints of e_ℓ (one in $V(\widehat{A}_{R_p}^{(C_j,q)})$ and the other in $I_q^{(C_j,R_p)}$) in an arbitrary order. When we reach any iteration corresponding to some $\ell \in [s]$, observe that the sets of vertices (from $D_{p,q}^j$) inserted in the previous iterations of this innermost loop form three modules (that are each either a clique or an independent set)—one consisting of the vertices inserted from $V(\widehat{A}_{R_p}^{(C_j,q)})$, another consisting of the vertices inserted from $I_q^{(C_j,R_p)}$, and the last consisting of the vertices inserted from $\widehat{M}_{R_p}^{(C_j,q)}$. In addition, each of the vertices inserted in the iteration corresponding to ℓ can form its own module before the iteration finishes, but only two such vertices are inserted (the third vertex finished the iteration). Overall, we derive that for all $i \in [n']$, $|EQ(G, \sigma, i)| \leq 2k + 24$.

By the arguments above, we have that $\text{nw}(G, \sigma) \leq 2k + 24$. Therefore, $\text{nw}(G) \leq 2k + O(1)$. This completes the proof. \square

We are now ready to conclude the correctness of our first main theorem.

THEOREM 3. *Unless ETH fails, GRAPH COLORING cannot be solved in time $f(k) \cdot n^{2^{o(k)}}$ for any function f of k , where k is the neighborhood-width of G .*

PROOF. Suppose, by way of contradiction, that there exists an algorithm \mathcal{A} that solves GRAPH COLORING in time $f(k) \cdot n^{2^{o(k)}}$ for some function f of k , where k is the neighborhood-width. As we would like to reserve n and k to be used in the context of 4-MONOTONE MIN-CSP, let us next use n' and k' , respectively, in the context of GRAPH COLORING, e.g., under this notation \mathcal{A} runs in time $f(k') \cdot n'^{2^{o(k'')}}$. Now, consider the following algorithm \mathcal{B} for 4-MONOTONE MIN-CSP. Given an instance (X, C, n, W) of 4-MONOTONE MIN-CSP where $k = 2\widehat{k}$, algorithm \mathcal{B} first constructs the instance $\text{red}(X, C, n, W) = (G, k')$ of GRAPH COLORING in polynomial time. Then, it calls algorithm \mathcal{A} with (G, k') as input, and answers Yes if the reply given by algorithm \mathcal{A} is at most η , and No otherwise. By Lemmata 3.4 and 3.10, algorithm \mathcal{B} is correct. Furthermore, in the output instance

$n' = n^{O(1)}$, and by Lemma 3.11, we also have that $k' = 2\widehat{k} + O(1) = 2\log_2 k + O(1)$. Thus, algorithm \mathcal{B} solves 4-MONOTONE MIN-CSP in time $f(k') \cdot n'^{2^{O(k')}} = f(\log k) \cdot (n^{O(1)})^{2^{O(\log k)}} = g(k)n^{o(k)}$ for some function g of k , which contradicts Lemma 3.1. This concludes the proof. \square

We remark that the proof above, in fact, shows that unless ETH fails, GRAPH COLORING cannot be solved in time $f(k) \cdot n^{o(2^{k/2})}$ for any function f of k , where k is the neighborhood-width of G . By Proposition 2.1, we have the following corollary to Theorem 3.

COROLLARY 3.1. *Unless ETH fails, GRAPH COLORING cannot be solved in time $f(k) \cdot n^{2^{o(k)}}$ for any function f of k , where k is the linear clique-width of G .*

Finally, due to Observation 2.1, Theorem 1 follows as a consequence of Corollary 3.1.

4 HAMILTONIAN CYCLE

In this section, we consider the HAMILTONIAN CYCLE problem. Recall that in HAMILTONIAN CYCLE, we are given a graph G and the objective is to check whether there exists a cycle passing through every vertex of G , i.e., a Hamiltonian cycle. We prove Theorem 2, which provides an algorithmic lower bound for the HAMILTONIAN CYCLE problem when parameterized by the clique-width of the input graph. To prove it, we give a reduction from the RED-BLUE CAPACITATED DOMINATING SET problem parameterized by the feedback vertex number of the input graph. Respectively, in Section 4.1, we introduce this problem, and in Section 4.2 give the proof of Theorem 2.

4.1 Capacitated Domination

A *red-blue capacitated graph* is a pair (G, c) , where G is a bipartite graph with the vertex bipartition R and B , and $c : R \rightarrow \mathbb{N}$ is a *capacity function* such that $1 \leq c(v) \leq d_G(v)$ for every vertex $v \in R$. The vertices of R are called *red* and the vertices of B are called *blue*. A set $S \subseteq R$ is called a *capacitated dominating set* if there is a *domination mapping* $f : B \rightarrow S$ mapping every vertex from B to one of its neighbors in S such that the total number of vertices mapped by f to each vertex $v \in S$ does not exceed its capacity $c(v)$. For $v \in S$, we say that vertices in $f^{-1}(v)$ are *dominated* by v . We consider the following variant of the CAPACITATED DOMINATION problem.

RED-BLUE CAPACITATED DOMINATING SET (RED-BLUE CDS)

Parameter: k

Input: A red-blue capacitated graph (G, c) .

Question: Does G contain a capacitated dominating set of size at most k ?

The investigation of the parameterized complexity of the CAPACITATED DOMINATION problem was initiated by Dom et al. [19] and Bodlaender, Lokshtanov, and Penninkx [5]. Further, RED-BLUE CDS proved to be a good tool problem for establishing hardness of problems parameterized by the clique-width of the input graph. In particular, some of the authors of this article used reductions from RED-BLUE CDS to show the W[1]-hardness of EDGE DOMINATING SET and HAMILTONIAN CYCLE in Ref. [23]; and then in Ref. [24], RED-BLUE CDS was used to establish asymptotically tight lower bounds for EDGE DOMINATING SET and MAX-CUT, assuming ETH (see also Ref. [7] for the related results). The following lemma was shown in Ref. [24].

LEMMA 4.1 ([24, THEOREM 3.1]). *Unless the ETH fails, RED-BLUE CDS cannot be solved in time $f(h)n^{o(h)}$, where h is the feedback vertex number of the input graph. Moreover, the problem cannot be solved in time $f(h)n^{o(h)}$ even if the input is restricted to graphs G such that for every minimum feedback vertex set $X \subseteq V(G)$,*

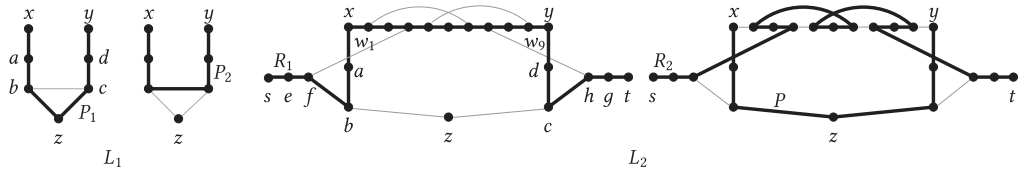


Fig. 5. Graphs L_1 and L_2 . Paths $P_1, P_2, R_1, R_2,$ and P are shown by thick lines.

- X is independent, and
- each vertex of the forest $G - X$ is adjacent to at most one vertex of X .

We also need the following upper bound for clique-width.

LEMMA 4.2 ([24, LEMMA 2.1]). *Let X be a feedback vertex set of a graph G such that each vertex v of the forest $F = G - X$ is adjacent to at most one vertex of X . Then $\text{cw}(G) \leq 4 \cdot |X| + 3$.*

4.2 Proof of Theorem 2

Now we are ready to prove the main results of the section. We reduce from RED-BLUE CDS parameterized by the feedback vertex number of the input graph. We first give a construction, then prove its correctness, and, finally, argue on the clique-width of the transformed instance.

Our reduction is, in fact, a variant of the reduction that was used in Ref. [23] to prove the $W[1]$ -hardness of HAMILTONIAN CYCLE parameterized by the clique-width of the input graph. The main difference is that now we have to construct a graph whose clique-width is linear in the feedback vertex number of the input graph. Hence, we modify the reduction to ensure this property and show the new upper bound for the clique-width.

As our reduction is based on the reduction from Ref. [23], we use some auxiliary results obtained in that article.

Auxiliary Gadgets. We denote, by L_1 , the graph with the vertex set $\{x, y, z, a, b, c, d\}$ and the edge set $\{xa, ab, bc, cd, dy, bz, cz\}$. Let P_1 be the path $xabzcdy$, and $P_2 = xabcdy$. (See Figure 5.)

We use the following property of this graph.

LEMMA 4.3 ([23, LEMMA 8]). *Let G be a Hamiltonian graph such that $G[V']$ is isomorphic to L_1 for $V' \subseteq V(G)$. Furthermore, if all edges in $E(G) \setminus E(G[V'])$ incident to V' are incident to the copies of the vertices $x, y,$ and z in V' , then every Hamiltonian cycle in G either includes the path P_1 , or the path P_2 as a segment.*

Our second auxiliary gadget is the graph L_2 . This graph has $\{x, y, z, s, t, a, b, c, d, e, f, g, h\}$ as its vertex set. We first include $xa, ab, bz, cz, cd, dy, se, ef, fb, ch, hg, gt$ in its edge set. Then, an (x, y) -path of length 10, $xw_1 \cdots w_9y$, is added, and edges $fw_3, w_1w_6, w_4w_9, w_7h$ are included in the set of edges. Let $P = xabzcdy$, $R_1 = sefbaxw_1w_2 \cdots w_9ydc hgt$, and $R_2 = sefw_3w_2w_1w_6w_5w_4w_9w_8w_7hgt$. (See Figure 5.) This graph has the following property.

LEMMA 4.4 ([23, LEMMA 8]). *Let G be a Hamiltonian graph such that $G[V']$ is isomorphic to L_2 for $V' \subseteq V(G)$. Furthermore, if all edges in $E(G) \setminus E(G[V'])$ incident to V' are incident to the copies of the vertices x, y, z, s, t in V' , then every Hamiltonian cycle in G includes either the path R_1 , or two paths P and R_2 as segments.*

Final Reduction. Now we describe our reduction. Let (G, c) be red-blue capacitated graph with $R = \{u_1, \dots, u_n\}$ being the set of red vertices and $B = \{v_1, \dots, v_r\}$ being the set of blue vertices and k be a positive integer.

The general idea of the reduction is to replace each red vertex u_i and the edges incident to u_i by gadgets to achieve the following property: if u_i is selected to be in a capacitated dominating

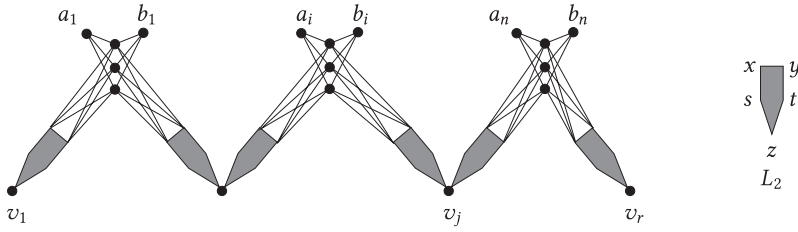


Fig. 6. Graph $G'(c)$.

set, then exactly $c(u_i)$ paths with end-vertices in the gadget corresponding to u_i and the internal vertices in the gadgets corresponding to the edges incident to u_i should form segments of a (potential) Hamiltonian cycle. Then, the property that a blue vertex v_j is dominated by u_i corresponds to the property that v_i is included in one of the paths. To achieve this, each red vertex u_i is replaced by two vertices a_i, b_i , the vertices a_i and b_i are joined by $c(u_i) + 1$ paths of length two. Let C_i denote the set of middle vertices of these paths, and $X_i = C_i \cup \{a_i, b_i\}$. Each edge $u_i v_j \in E(G)$, is replaced by a copy L_2^{ij} of L_2 with $z = v_j$ and vertices x and y are made adjacent to all the vertices of C_i . The vertices corresponding to s and t are called s_{ij} and t_{ij} in L_2^{ij} . Furthermore, let x_{ij} and y_{ij} denote the vertices corresponding to x and y in L_2^{ij} . The paths corresponding to P, R_1 , and R_2 are called P^{ij}, R_1^{ij} , and R_2^{ij} , respectively, in L_2^{ij} . Denote the obtained graph by $G'(c)$. (See Figure 6 for an illustration.)

We are going to have k red vertices selected to be in a capacitated dominating set. Respectively, $n - k$ red vertices should be outside of the set. Then the vertices of the gadgets constructed above for these vertices and the incident edges should be included in a Hamiltonian cycle (if exists). Therefore, we need a gadget that allows to constrict segments of a Hamiltonian cycle to collect these vertices. We add two vertices g and h , which are joined by $\sum_{i=1}^n (c(v_i) + 3) + m + 1$ paths of length two where $m = |E(G)|$. Let Y be the set of middle vertices of these paths. All vertices s_{ij} and t_{ij} are joined by edges with all vertices of Y . For every vertex w such that $w \in X_i$ (recall $X_i = C_i \cup \{a_i, b_i\}$), $i \in [n]$, a copy L_1^w of L_1 with $z = w$ is attached and the vertices x, y of this gadget are joined to all vertices of Y . We let x_w and y_w denote the vertices corresponding to x and y in L_1^w . Similarly, P_1^w and P_2^w denotes the paths in L_1^w corresponding to P_1 and P_2 , respectively.

Finally, we need a selection gadget constructed by adding $k + 1$ vertices, namely p_1, \dots, p_{k+1} , and making them adjacent to the vertices of $\{a_i, b_i : 1 \leq i \leq n\}$ and to g and h . The segments of a (potential) Hamiltonian cycle with their end-vertices in $\{p_1, \dots, p_{k+1}\}$ are used to select k gadgets corresponding to the red vertices that are going to be included in a capacitated dominating set.

Denote by H the obtained graph. The construction of H can easily be done in time polynomial in n and r .

LEMMA 4.5. *A graph (G, c) has a capacitated dominating set of size at most k if and only if H has a Hamiltonian cycle.*

PROOF. Let S be a capacitated dominating set of size at most k in (G, c) with the corresponding dominating mapping f . Without loss of a generality, we assume that $|S| = k$ and $S = \{u_1, \dots, u_k\}$. The Hamiltonian cycle we are trying to construct is naturally divided into $k + 1$ parts by the vertices p_1, \dots, p_{k+1} . We construct the Hamiltonian cycle starting from the vertex p_1 . Assume that the part of the cycle up to the vertex p_i is already constructed. We show how to construct the part from p_i to p_{i+1} . We include the edge $p_i a_i$ in it. Let $J = \{j \in \{1, \dots, x\} : f(v_j) = u_i\}$. If $J = \emptyset$, then a_i is joined with b_i by a path of length two, which goes through one vertex of C_i . Otherwise, for all gadgets L_2^{ij} where $j \in J$, the paths P^{ij} are included to the cycle as segments, and endpoints of

these paths are joined consecutively by paths of length two through vertices of C_i with a_i and b_i (that is, a_i is joined with one endpoint of the first path through a vertex of C_i , another endpoint of this path is joined the endpoint of the second path through another vertex of C_i , and so on; the remaining endpoint of the last path is joined with b_i). Since $|J| \leq c(u_i)$ and $|C_i| = c(u_i)$, we can always find vertices in C_i for this construction. Finally we include the edge $b_i p_{i+1}$ to the cycle.

When the vertex p_{k+1} is reached, we move to the set Y . Note that, at this stage, the vertices v_1, \dots, v_r are already included in the cycle. We start by including the edge $p_{k+1}g$. We will add the following segments to the cycle and connect them appropriately.

- For every L_2^{ij} , the path R_1^{ij} is added to the cycle if P^{ij} was not included to it; else, the path R_2^{ij} is added. Note that m such paths are included to the cycle.
- For every vertex w such that $w \in X_i$ for some $i \in [n]$, the path P_2^w is included in the cycle if w is already included in the constructed part of the cycle; else the path P_1^w is added. Clearly, we add $\sum_{i=1}^n (c(v_i) + 3)$ paths.

Finally, the total number of paths we will add is $\sum_{i=1}^n (c(v_i) + 3) + m = |Y| - 1$. We add the segments of the paths mentioned with the help of vertices in Y in the way we added the paths P^{ij} with the help of vertices in C_i . Let the end points of the resultant joined path be $\{q_1, q_2\}$. Notice that (a) $q_1, q_2 \in Y$ and (b) this path includes all the vertices of Y . Now, we add edges gq_1, q_2h , and hp_1 . This completes the construction of the Hamiltonian cycle.

For the reverse direction of the proof, we assume that we have been given C , a Hamiltonian cycle in H . This cycle is divided into $k + 1$ segments by the vertices p_1, \dots, p_{k+1} . Let $S = \{u_i : p_j a_i \in E(C), a_i p_s \notin E(C), j \neq s, \text{ for some } j \in [k + 1]\}$. We prove that S is a capacitated dominating set in G of cardinality at most k . We first argue about the size of S ; clearly its size is upper bounded by $k + 1$. To argue that it is at most k , it is enough to observe that, by Lemmata 4.3 and 4.4, either $p_j g$ or $p_j h$ must be in $E(C)$ for some $j \in [k + 1]$. Now, we show that S is indeed a capacitated dominating set. Our proof is based on the following observations.

- By Lemma 4.4, every vertex v_j appears in a segment P^{ij} for some $j \in [r]$ in C . We set the domination function $f(v_j) = u_i$ if v_j is included in the segment P^{ij} in C .
- By Lemmata 4.3 and 4.4, the endpoints of paths P^{ij} can be reached only through vertices a_i and b_i from outside of the set X_i . This implies that all paths P^{ij} , which appear as segments in C for some $i \in [n]$, are joined together and with vertices a_i and b_i into one segment of C by paths that go through vertices of C_i . It means that $u_i \in S$ and $f(B) \subseteq S$. Moreover, since $|C_i| = c(u_i) + 1$, at most $c(u_i)$ paths P^{ij} can be segments of C for each $i \in [n]$ and, therefore, $|f^{-1}(u_i)| \leq c(u_i)$ for $u_i \in S$.

This concludes the proof of the lemma. □

The next lemma upper bounds the clique-width of H .]

LEMMA 4.6. *Let G be such that there is a minimum feedback vertex set $X \subseteq V(G)$ such that (i) X is independent, and (ii) each vertex of the forest $G - X$ is adjacent to at most one vertex of X . Then, $\text{cw}(H) \leq 16 \cdot |X| + 36$.*

PROOF. Let \hat{G} be the graph obtained from G by subdividing each edge of G , that is, for each edge $e \in E(G)$, we replace e by a new vertex w_e and make it adjacent to the end-vertices of e ; we say that w_e is the e -vertex of \hat{G} .

Clearly, X is a feedback vertex set of \hat{G} . Because of (i) and (ii), every vertex of $\hat{G} - X$ is adjacent to at most one vertex of X . By Lemma 4.2, $\text{cw}(\hat{G}) \leq 4 \cdot |X| + 3$. Let $t = 4 \cdot |X| + 3$. Consider an expression tree T of \hat{G} of width t and assume that the labels from the set $[t]$ are used in the

construction of G with respect to T . We construct the expression tree T^* for H of width $(4t + 24)$ by making modifications of T .

We use the following five groups of disjoint labels:

- Labels $\alpha_1, \dots, \alpha_r$ for the vertices of sets C_i for $i \in [n]$.
- Labels β_1, \dots, β_t for the vertices v_1, \dots, v_r .
- Labels $\gamma_1, \dots, \gamma_t$ and $\delta_1, \dots, \delta_t$ for the vertices in the copies of L_2 , which are adjacent to the vertices of C_1, \dots, C_n and v_1, \dots, v_r , respectively.
- Labels ξ_1, ξ_2, ξ_3 for marking some vertices.
- Working labels $\zeta_1, \dots, \zeta_{21}$.

First, we consider the leaves of T where the vertices of \hat{G} are introduced and replace these introduce nodes by expression trees for induced subgraphs of H .

Let $p(w_e)$ be an introduce node for an e -vertex of \hat{G} where $e = u_i v_j$ for some $i \in [n]$ and $j \in [r]$. We use the labels $\zeta_1, \dots, \zeta_{21}$ to create the expression tree for a copy of the graph $L_2 - z$ (for simplicity, we use a separate label for each vertex). Then, we add the path of relabel nodes whose one end-node is made adjacent to the root and the other end-node becomes the new root. The relabel nodes are used to relabel the vertices x and y of this copy of $L_2 - z$ by γ_p , the vertices b and c (adjacent to z) by δ_p , the vertices s and t by ξ_1 , and the remaining vertices are relabeled by ξ_3 .

Let $p(u_i)$ be an introduce node for u_i , $i \in [n]$. We construct the expression tree for the graph obtained as follows by making use of introduce, disjoint union, relabel, and join operations. We omit the union operations from our descriptions here and from the forthcoming descriptions of this type assuming, implicitly, that if some vertex is introduced, then union is always performed. First, we create the vertex a_i labeled ζ_1 . Then, the six vertices of a copy of the gadget L_1 attached to a_i labeled $\zeta_5, \dots, \zeta_{10}$ are introduced, and edges of L_1 are created by corresponding join operations. We relabel vertices x and y of this copy of L_1 by ξ_1 and remaining vertices of $L_1 - z$ are relabeled by ξ_3 . Next step is to introduce b_i with the label ζ_2 . After this, we introduce a copy of L_1 attached to b_i , relabel x and y by ξ_1 and relabel remaining vertices of $L_1 - z$ by ξ_3 . Now, we repeat the following $c(u_i)$ times (to create vertices of C_i with attached gadgets L_1): introduce a vertex labeled ζ_3 , use the labels $\zeta_5, \dots, \zeta_{10}$ (together with the vertex labeled ζ_3) to make a copy of L_1 , relabel x and y of L_1 by ξ_1 , relabel vertices $L_1 - z$ by ξ_3 , and relabel the vertex labeled ζ_3 by ζ_4 . Finally, the vertices labeled by ζ_4 are joined with the vertices labeled ζ_1 and ζ_2 ; the vertices a_i and b_i are relabeled by ξ_2 , and the vertices labeled by ζ_4 are relabeled by α_p .

Finally, let $p(v_j)$ be an introduce node for v_j , $j \in [r]$. In this case, we replace this node by the introduce node $\beta_p(v_j)$.

Now, we consider non-leaf nodes of T . For every such node Z , we assume that the new expression trees corresponding to the subtrees of T rooted in the children of Z are already constructed.

If $Z = \oplus$, then we keep the union node whose children are the roots of the expression trees constructed for the children of Z in T .

Let $Z = \eta_{p,q}$ for $p, q \in [t]$. We create the path of join nodes $\eta_{\alpha_p, \gamma_q}$, η_{δ_p, β_q} , $\eta_{\alpha_q, \gamma_p}$, and η_{δ_q, β_p} . One end-node is made the parent for the root of the expression tree that is constructed for the child of Z in T , and the other end-node of the path is the root of the expression tree for Z .

Let $Z = \rho_{p \rightarrow q}$ for $p, q \in [t]$. We construct the path of relabel nodes $\rho_{\alpha_p \rightarrow \alpha_q}$, $\rho_{\beta_p \rightarrow \beta_q}$, $\rho_{\gamma_p \rightarrow \gamma_q}$, and $\rho_{\delta_p \rightarrow \delta_q}$. In the same way as above, one end-node is made the parent for the root of the expression tree that is constructed for the child of Z in T , and the other end-node of the path is the root of the expression tree for Z .

This completes the part of the construction of T^* where we followed T starting from the leaves. Denote by T' the expression tree obtained in this stage.

We construct the expression tree for the following graph. We construct vertices g and h using labels ζ_1 and ζ_2 . Then, $\sum_{i=1}^n (c(v_i) + 3) + m + 1$ vertices of Y labeled ζ_3 are introduced and joined with the vertices labeled ζ_1 and ζ_2 . The vertices g, h are relabeled by ζ_2 . We construct the union node with the children in the roots of this expression tree and T' . Notice that all the vertices that have to be joined with vertices of Y are labeled by ζ_1 . So, we construct the join node η_{ζ_3, ζ_1} that is the parent of the union node constructed above. Let T'' be the obtained expression tree.

To complete the construction of T^* , it remains to construct the vertices p_1, \dots, p_{k+1} and connect them with the already constructed part of H . Notice that these vertices should be made adjacent to the vertices labeled ζ_2 . To do this, we construct $k + 1$ introduce nodes $\zeta_4(p_i)$ for $i \in [k + 1]$. Then, we construct $k + 1$ union nodes that are used to make the disjoint union of p_1, \dots, p_{k+1} and the graph corresponding to the root of T'' . Finally, we construct a join node η_{ζ_2, ζ_4} .

Following the steps of the construction of T^* , it is straightforward to see that the graph associated with the root of T^* is, indeed, H . Since the width of T^* is $4t + 24$, we have that $\text{cw}(H) \leq 4t + 24 = 16 \cdot |X| + 36$. \square

Now, we are ready to complete the proof of Theorem 2. Recall that by Lemma 4.1, unless ETH fails, RED-BLUE CDS cannot be solved in time $f(h)n^{o(h)}$, where h is the feedback vertex number of the input graph, even if the input is restricted to graphs G such that for every minimum feedback vertex set $X \subseteq V(G)$, (i) X is independent and (ii) each vertex of the forest $G - X$ is adjacent to at most one vertex of X . Given such an input (G, c) of RED-BLUE CDS, we construct the graph H . By Lemma 4.5, (G, c) is a yes-instance of RED-BLUE CDS if and only if H is Hamiltonian; that is, H is a yes-instance of HAMILTONIAN CYCLE. The construction of H can be done in polynomial time. By Lemma 4.6, $\text{cw}(H) \leq 16 \cdot \text{fvn}(G) + 36$. This implies that HAMILTONIAN CYCLE cannot be solved in time $f(t)n^{o(t)}$, where t is the clique-width of the input graph, unless ETH fails.

5 CONCLUSION

In this article, we proved that unless the ETH fails, GRAPH COLORING cannot be solved in time $O(f(k) \cdot n^{2^{o(k)}})$ and HAMILTONIAN CYCLE cannot be solved in time $O(f(k) \cdot n^{o(k)})$, where k is the clique-width of the input graph. At this point, the complexity of MAX-CUT, EDGE DOMINATING SET, GRAPH COLORING, and HAMILTONIAN CYCLE on graphs of bounded clique-width is quite well-understood. On the other hand, pinning down the right exponent of n for these problems on graphs of rank-width k remains open. The more intriguing open problem remains the complexity of computing the clique-width of a graph. To the best of our knowledge, it is consistent with current knowledge that determining whether G has clique-width k is FPT or that determining whether G has clique-width k has an algorithm with running time $g(k) \cdot n^{f(k)}$ and is W[1]-hard parameterized by k , or that determining whether G has clique-width k is NP-complete for some fixed constant $k \geq 5$.

ACKNOWLEDGMENTS

We thank Michał Pilipczuk for insightful discussions.

REFERENCES

- [1] Stefan Arnborg, Jens Lagergren, and Detlef Seese. 1991. Easy problems for tree-decomposable graphs. *J. Algorithms* 12, 2 (1991), 308–340.
- [2] Benjamin Bergougnoux, Mamadou Moustapha Kanté, and O-joung Kwon. 2017. An optimal XP algorithm for Hamiltonian cycle on graphs of bounded clique-width. In *Proceedings of the 15th International Symposium on Algorithms and Data Structures (Lecture Notes in Computer Science)*, Vol. 10389. Springer, 121–132.
- [3] Umberto Bertelé and Francesco Brioschi. 1972. *Nonserial Dynamic Programming*. Academic Press.
- [4] Hans L. Bodlaender. 1998. A partial k -arboretum of graphs with bounded treewidth. *Theoret. Comput. Sci.* 209, 1–2 (1998), 1–45.

- [5] Hans L. Bodlaender, Daniel Lokshtanov, and Eelko Penninkx. 2009. Planar capacitated dominating set is $W[1]$ -hard. In *Proceedings of the 4th Fourth International Workshop on Parameterized and Exact Computation (IWPEC'09) (Lecture Notes in Computer Science)*, Vol. 5917. Springer, 50–60. DOI:https://doi.org/10.1007/978-3-642-11269-0_4
- [6] Richard B. Borie, R. Gary Parker, and Craig A. Tovey. 1992. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica* 7, 5–6 (1992), 555–581.
- [7] Hajo Broersma, Petr A. Golovach, and Viresh Patel. 2013. Tight complexity bounds for FPT subgraph problems parameterized by the clique-width. *Theor. Comput. Sci.* 485 (2013), 69–84.
- [8] Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. 2011. Boolean-width of graphs. *Theor. Comput. Sci.* 412, 39 (2011), 5187–5204.
- [9] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. 2009. The complexity of satisfiability of small depth circuits. In *Proceedings of the 4th International Workshop on Parameterized and Exact Computation (Lecture Notes in Computer Science)*, Vol. 5917. Springer, 75–85.
- [10] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. 2006. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.* 72, 8 (2006), 1346–1367.
- [11] Derek G. Corneil, Michel Habib, Jean-Marc Lanlignel, Bruce A. Reed, and Udi Rotics. 2012. Polynomial-time recognition of clique-width ≤ 3 graphs. *Discrete Appl. Math.* 160, 6 (2012), 834–865.
- [12] Derek G. Corneil and Udi Rotics. 2005. On the relationship between clique-width and treewidth. *SIAM J. Comput.* 34, 4 (2005), 825–847.
- [13] B. Courcelle. 1992. The monadic second-order logic of graphs. III. Tree-decompositions, minors and complexity issues. *RAIRO Inform. Théor. Appl.* 26, 3 (1992), 257–286.
- [14] B. Courcelle, J. A. Makowsky, and U. Rotics. 2000. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.* 33, 2 (2000), 125–150.
- [15] Bruno Courcelle and Stephan Olariu. 2000. Upper bounds to the clique width of graphs. *Discrete Appl. Math.* 101, 1–3 (2000), 77–114.
- [16] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer. DOI:<https://doi.org/10.1007/978-3-319-21275-3>
- [17] Marek Cygan, Marcin Pilipczuk, and Michal Pilipczuk. 2016. Known algorithms for edge clique cover are probably optimal. *SIAM J. Comput.* 45, 1 (2016), 67–83.
- [18] Reinhard Diestel. 2010. *Graph Theory* (4th ed.). Springer-Verlag, Heidelberg, Germany.
- [19] Michael Dom, Daniel Lokshtanov, Saket Saurabh, and Yngve Villanger. 2008. Capacitated domination and covering: A parameterized perspective. In *IWPEC 2008 (Lecture Notes in Computer Science)*, Vol. 5018. Springer, 78–90. DOI:https://doi.org/10.1007/978-3-540-79723-4_9
- [20] Rodney G. Downey and Michael R. Fellows. 2013. *Fundamentals of Parameterized Complexity*. Springer. DOI:<https://doi.org/10.1007/978-1-4471-5559-1>
- [21] Wolfgang Espelage, Frank Gurski, and Egon Wanke. 2001. How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time. In *Proceedings of the 27th International Workshop on Graph-Theoretic Concepts in Computer Science, WG (Lecture Notes in Computer Science)*, Vol. 2204. Springer, 117–128.
- [22] Michael R. Fellows, Frances A. Rosamond, Udi Rotics, and Stefan Szeider. 2009. Clique-width is NP-complete. *SIAM J. Discrete Math.* 23, 2 (2009), 909–939.
- [23] Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. 2010. Intractability of clique-width parameterizations. *SIAM J. Comput.* 39, 5 (2010), 1941–1956.
- [24] Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. 2014. Almost optimal lower bounds for problems parameterized by clique-width. *SIAM J. Comput.* 43, 5 (2014), 1541–1563.
- [25] Markus Frick and Martin Grohe. 2004. The complexity of first-order and monadic second-order logic revisited. *Ann. Pure Appl. Logic* 130, 1–3 (2004), 3–31.
- [26] Michael U. Gerber and Daniel Kobler. 2003. Algorithms for vertex-partitioning problems on graphs with fixed clique-width. *Theoret. Comput. Sci.* 299, 1–3 (2003), 719–734.
- [27] Omer Giménez, Petr Hliněný, and Marc Noy. 2006. Computing the tutte polynomial on graphs of bounded clique-width. *SIAM J. Discret. Math.* 20, 4 (2006).
- [28] Benny Godlin, Tomer Kotek, and Johann A. Makowsky. 2008. Evaluations of graph polynomials. In *Proceedings of the 34th International Workshop on Graph-Theoretic Concepts in Computer Science (Lecture Notes in Computer Science)*, Vol. 5344. 183–194.
- [29] Frank Gurski. 2006. Linear layouts measuring neighbourhoods in graphs. *Discrete Math.* 306, 15 (2006), 1637–1650. DOI:<https://doi.org/10.1016/j.disc.2006.03.048>
- [30] Rudolf Halin. 1976. S-functions for graphs. *J. Geom.* 8, 1 (1976), 171–186.

- [31] Petr Hliněný and Sang-il Oum. 2008. Finding branch-decompositions and rank-decompositions. *SIAM J. Comput.* 38, 3 (2008), 1012–1032.
- [32] Petr Hliněný, S.-il Oum, Detlef Seese, and Georg Gottlob. 2008. Width parameters beyond tree-width and their applications. *Comput. J.* 51, 3 (2008), 326–362.
- [33] Russell Impagliazzo and Ramamohan Paturi. 2001. On the complexity of k -SAT. *J. Comput. Syst. Sci.* 62, 2 (2001), 367–375. DOI : <https://doi.org/10.1006/jcss.2000.1727>
- [34] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. 2001. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.* 63, 4 (2001), 512–530. DOI : <https://doi.org/10.1006/jcss.2001.1774>
- [35] Daniel Kobler and Udi Rotics. 2001. Polynomial algorithms for partitioning problems on graphs with fixed clique-width (extended abstract). In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*. 468–476.
- [36] Daniel Kobler and Udi Rotics. 2003. Edge dominating set and colorings on graphs with fixed clique-width. *Discrete Appl. Math.* 126, 2–3 (2003), 197–221.
- [37] Michael Lampis. 2014. Model checking lower bounds for simple graphs. *Logical Methods Comput. Sci.* 10, 1 (2014), 1–21.
- [38] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. 2011. Lower bounds based on the exponential time hypothesis. *Bull. EATCS* 105 (2011), 41–72. DOI : [cs/article/view/96](https://doi.org/10.1007/s00037-011-0096-9)
- [39] Johann A. Makowsky, Udi Rotics, Ilya Averbouch, and Benny Godlin. 2006. Computing graph polynomials on graphs of bounded clique-width. In *Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science (Lecture Notes in Computer Science)*, Vol. 4271. Springer, 191–204.
- [40] Dániel Marx and Valia Mitsou. 2016. Double-exponential and triple-exponential bounds for choosability problems parameterized by treewidth. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming*, Vol. 55. 28:1–28:15.
- [41] S.-il Oum and Paul Seymour. 2006. Approximating clique-width and branch-width. *J. Combin. Theory Ser. B* 96, 4 (2006), 514–528.
- [42] Sang-il Oum. 2008. Approximating rank-width and clique-width quickly. *ACM Trans. Algorithms* 5, 1 (2008), 10:1–10:20.
- [43] Michaël Rao. 2007. MSOL partitioning problems on graphs of bounded treewidth and clique-width. *Theoret. Comput. Sci.* 377, 1–3 (2007), 260–267.
- [44] Neil Robertson and Paul D. Seymour. 1984. Graph minors. III. planar tree-width. *J. Comb. Theory, Ser. B* 36, 1 (1984), 49–64.
- [45] Karol Suchan and Ioan Todinca. 2007. On powers of graphs of bounded NLC-width (clique-width). *Discrete Appl. Math.* 155, 14 (2007), 1885–1893.
- [46] Egon Wanke. 1994. k -NLC graphs and polynomial algorithms. *Discrete Appl. Math.* 54, 2–3 (1994), 251–266.

Received February 2018; revised September 2018; accepted September 2018