Cliquewidth III: The Odd Case of Graph Coloring Parameterized by Cliquewidth

Petr A. Golovach* Daniel Lokshtanov† Saket Saurabh‡ Meirav Zehavi§

Abstract

Max-Cut (MC), Edge Dominating Set (EDS), Graph Coloring (GC) and Hamiltonian Path (HP) on graphs of bounded cliquewidth 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 cliquewidth 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 cliquewidth k. Fomin et al. [*Intractability of Clique-Width Parameterizations. SIAM J. Comput. 39(5): 1941-1956 (2010)*] showed that the running times cannot be improved to $g(k)n^{\mathcal{O}(1)}$ assuming $\mathsf{W[1]}\neq\mathsf{FPT}$. 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. [*Almost Optimal Lower Bounds for Problems Parameterized by Clique-Width. SIAM J. Comput. 43(5): 1541-1563 (2014)*] improved the running times for EDS and MC to $n^{\mathcal{O}(k)}$, and proved $g(k)n^{o(k)}$ lower bounds for EDS, MC and HP assuming the ETH. Recently, Bergougnoux, Kanté and Kwon [*WADS 2017*] gave an $n^{\mathcal{O}(k)}$-time algorithm for HP. Thus, prior to this work, EDS, MC and HP were known to have tight $n^{\Theta(k)}$ algorithmic upper and lower bounds. In contrast, GC has an upper bound of $n^{\mathcal{O}(2^k)}$ and a lower bound of merely $n^{o(\sqrt[4]{k})}$ (implicit from the $\mathsf{W[1]}$-hardness proof). In this paper, we close the gap for GC by proving a lower bound of $n^{2^{o(k)}}$. This shows that GC behaves qualitatively different from the other three problems. To the best of our knowledge, GC is the first natural problem known to require exponential dependence on the parameter in the exponent of n.

*Department of Informatics, University of Bergen, Norway. petr.golovach@uib.no

†University of Bergen, Bergen, Norway. daniello@ii.uib.no

‡University of Bergen, Bergen, Norway, and The Institute of Mathematical Sciences, HBNI, Chennai, India. saket@imsc.res.in

§University of Bergen, Bergen, Norway. meirav.zehavi@ii.uib.no

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 *cliquewidth*, 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 [27], and finally in 1984, Robertson and Seymour [41] 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 [13] defined a 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 *cliquewidth* of the graph. Cliquewidth is a generalization of treewidth in the sense that graphs of bounded treewidth also have bounded cliquewidth [10]. It is also worth to mention here the related graph parameters NLC-width, introduced by Wanke [43], rankwidth introduced by Seymour and Oum [39], and booleanwidth, introduced by Bui-Xuan, Telle and Vatshelle [6]. We refer to the survey of Hlinený et al. [29] for further references on cliquewidth and related parameters.

In the last decade, cliquewidth as a graph parameter has received significant attention. Corneil et al. [9] show that graphs of cliquewidth at most 3 can be recognized in polynomial time. Fellows et al. [19] settled a long standing open problem by showing that computing cliquewidth is NP-hard. Oum and Seymour [39] describe an algorithm that, for any fixed k, runs in time $\mathcal{O}(n^9 \log n)$ and computes $(2^{3k+2}-1)$-expressions for a n-vertex graph G of clique-width at most k. Oum [38] improved this result by providing an algorithm computing (8^k-1)-expressions in time $\mathcal{O}(n^3)$. Finally, Hliněný and Oum [28] obtained an algorithm running in time $\mathcal{O}(n^3)$ and computing $(2^{k+1}-1)$-expressions for a graph G of cliquewidth at most k.

Most of the algorithms on graphs of bounded treewidth and cliquewidth are based on dynamic programming over the corresponding decomposition tree, are very similar to each other. This similarity hinted at the existence of meta-theorems that could simultaneously provide algorithms on bounded treewidth and cliquewidth graphs for large classes of problems. Indeed, Courcelle [11] (see also [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. 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 [12] extended the meta-theorem of Courcelle to graphs of bounded cliquewidth. 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 cliquewidth k. Thus, these problems are FPT parameterized by the cliquewidth and the length of the formula.

Comparing the two meta-theorems reveals a trade-off between expressive power of the logic and applicability to larger (bounded cliqewidth) or smaller (bounded treewidth) classes of graphs. This leads to the question on whether this trade-off is unavoidable. Courcelle, Makowsky, and Rotics [12] 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 NEXP = EXP. For several natural graph problems, such as Max-Cut (MC), Edge Dominating Set (EDS),

Graph Coloring (GC) and Hamiltonian Path (HP), linear time algorithms on bounded treewidth graphs were known to follow from (variants of [1, 5]) Courcelle's theorem [11]. At the same time these problems, and many others, were known to admit algorithms with running time $\mathcal{O}(n^{f(k)})$ on graphs with n vertices and cliquewidth k [18, 23, 24, 25, 32, 33, 36, 40, 42, 43]. The existence of FPT algorithms (parameterized by the cliquewidth k of the input graph) for these problems (or their generalizations) was asked as open problems by Gerber and Kobler [23], Kobler and Rotics [32, 33], Makowsky, Rotics, Averbouch, Kotek, and Godlin [36, 25].

A subset of the authors of the current paper, together with Fomin [20], showed that the EDS, HP, and GC problems parameterized by cliquewidth are all W[1]-hard. In particular, this implies that these problems do not admit algorithms with running times of the form $\mathcal{O}(g(k)\cdot n^c)$, for any function g and constant c independent of k, unless FPT=W[1]. However, the lower bounds of Fomin et al. [20] did not rule out non-trivial improvements to the exponent $f(k)$ of n in the running times.

In a follow-up paper, Fomin et al. [21] improved the running times for EDS and MC from $n^{\mathcal{O}(k^2)}$ to $n^{\mathcal{O}(k)}$, and proved $g(k)n^{o(k)}$ lower bounds for EDS, MC and HP, assuming the Exponential Time Hypothesis (ETH). Together, these lower and upper bounds gave asymptotically tight algorithmic bounds for EDS and MC. However, the best known algorithm for HP ran in time $n^{\mathcal{O}(k^2)}$. Recently, Bergougnoux, Kanté and Kwon [2] gave a highly non-trivial $n^{\mathcal{O}(k)}$-time algorithm for HP. This together with the lower bound provided by Fomin et al. [21] for HP gives asymptotically tight algorithmic bound for HP. Thus, prior to this work, EDS, MC and HP were known to have tight $n^{\Theta(k)}$ algorithmic upper and lower bounds. In contrast, GC has an upper bound of $n^{\mathcal{O}(2^k)}$ [33] and a lower bound of merely $n^{o(\sqrt[4]{k})}$ (implicit from the W[1]-harness proof). This paper bridges this gap between the upper and lower bounds for GC by proving 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 $\mathcal{O}(f(k)\cdot n^{2^{o(k)}})$ for any function f of k, where k is the cliquewidth of G.*

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 cliquewidth* k, even when a linear cliquewidth expression (see [26]) of width at most k is given as input.

Theorem 1 shows that GC behaves qualitatively different from every other problem previously studied on graphs of bounded cliquewidth. Indeed, to the best of our knowledge, GC parameterized by cliquewidth 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 [15, 22, 34, 37]. However these lower bounds are all for the $g(k)$ factor of FPT algorithms, and not for the exponent of the input size n.

Methods. The key insights of our lower bound proof are in some sense dual to the key insights of the $n^{2^{O(k)}}$ time algorithm [33]. It is convenient to consider graphs of bounded *neighborhood-width* rather than bounded cliquewidth. In this setting the vertices of G are given according to an ordering $\sigma = v_1^\sigma, v_2^\sigma, \ldots, v_n^\sigma$, and satisfy the following property. For every $i \leq n$ the vertex set $\{v_1^\sigma, \ldots, v_i^\sigma\}$ can be partitioned into k sets $S_1, \ldots S_k$ such that the sets S_i 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 $\{v_{i+1}^\sigma, \ldots, v_n^\sigma\}$.

Consider a coloring algorithm that tries to color the the vertices of G in the order given by σ using at most η colors. When the vertices $\{v_1^\sigma, \ldots, v_i^\sigma\}$ have already been colored, this affects which colors can be used on the remaining vertices. For each color c the set of vertices in $\{v_{i+1}^\sigma, \ldots, v_n^\sigma\}$ that can *not* be colored by c are exactly the vertices that have at least one

neighbor in $\{v_1^\sigma, \ldots, v_i^\sigma\}$ colored with c. This vertex set is completely determined by the subset I_c of $\{1, \ldots, k\}$ 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 $\{v_{i+1}^\sigma, \ldots, v_n^\sigma\}$ 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 evey subset I of $\{1, \ldots, k\}$, the number of colors c such that $I_c = I$. This gives rise to a $n^{2^{\mathcal{O}(k)}}$ 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 $f(k)n^{o(k)}$ [8, 14] 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 $1, \ldots, n$ - 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 $S_1, \ldots S_k$ 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 4.

Overview. In Section 2 we set up basic notations and definitions. In Section 3 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 (Sections 3.1 and 3.2) 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-CSPto GC on graphs of bounded cliquewidth. This reduction is presented in Section 4. In Section 5 we wrap up with concluding remarks and open problems.

2 Preliminaries

We use $[n]$ and $[n]_0$ as shorthands for $\{1, 2, \ldots, n\}$ and $\{0, 1, \ldots, n\}$, respectively. Given a function $f : A \to B$, we let $\mathsf{dom}(f)$ and $\mathsf{ima}(f)$ 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 [16] 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. 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 *coloring* of G is a function $\chi : V(G) \to \mathbb{N}$. 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 $\chi : V(G) \to [t]$, that is, a proper coloring that uses only t colors.

Cliquewidth. 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 *cliquewidth* $\mathtt{cw}(G)$ 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*,

$\rho_{i\to j}$: *Relabel* operation changing all labels i to j, and

$\eta_{i,j}$: *Join* operation making all vertices labeled by i adjacent to all vertices labeled by j.

The *linear cliquewidth* $\mathtt{lcw}(G)$ 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 cliquewidth is more restrictive than the set of operations relevant to cliquewidth, the following observation is correct.

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

Let us now present an almost equivalent definition of linear cliquewidth, known as the *neighborhood width*. To this end, let σ be an ordering of $V(G)$ as $v_1^\sigma, v_2^\sigma, \ldots, v_n^\sigma$. For all $i \in [n]$, denote $V_i^\sigma = \{v_1^\sigma, v_2^\sigma, \ldots, v_i^\sigma\}$. 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 $EQ(G, \sigma, i)$, is the partition $\{S_1, S_2, \ldots, S_t\}$ 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.

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

The following proposition asserts that for our purpose, we can work with $\mathtt{nw}(G)$ rather than $\mathtt{lcw}(G)$.

Proposition 2.1 ([26])**.** *For any graph G, $\mathtt{lcw}(G) \leq \mathtt{nw}(G) + 1$.*

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 *fixed-parameter tractable (FPT)* if any instance (I, k) of Π is solvable in time $f(k) \cdot |I|^{\mathcal{O}(1)}$, 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 *Exponential-Time Hypothesis (ETH)* [30, 31, 7]. 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 $\mathcal{O}(2^{o(n)})$. Additional details on Parameterized Complexity and ETH can be found in [17, 14].

3 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 $\chi : V(G) \to [k]$.	
Question: Does G contain a multicolored clique C on k vertices?	

For Multicolored Clique, we have the following known proposition.

Proposition 3.1 ([35])**.** *Unless ETH fails,* Multicolored Clique *cannot be solved in time* $\mathcal{O}(f(k) \cdot n^{o(k)})$ *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 4, 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 $\alpha : X \to [n]_0$ is called an *assignment*. The *cost* of an assignment α, denoted by $\texttt{cost}(\alpha)$, is $\sum_{x \in X} \alpha(x)$. Given $X' \subseteq X$, a set R of pairs (x, c) such that $x \in X'$ and $c \in [n]$ 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 $C = (X', \mathcal{R})$ for some $m \in \mathbb{N}$, 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 $\mathcal{C}$ of constraints if α satisfies *every* constraint in $\mathcal{C}$.

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

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* $\mathcal{O}(f(k) \cdot n^{o(k)})$ *for any function* f *of* k*.*

3.1 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 $\chi^{-1}(i) = \{v_1^i, v_2^i, \ldots, v_{n'}^i\}$.

Let us now construct an instance $\texttt{red}(G, \chi, k) = (X, \mathcal{C}, n', W)$ 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 $X = \{x_1, x_2, \ldots, x_k\} \cup \{\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_k\}$ 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^i_j as the vertex of color i selected into a multicolored clique of size k. We will force the copy $\overline{x}_i$ 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 $\overline{x}_i$. Moreover, we define $W = k(n'+1)$. Now, it remains to define the set $\mathcal{C}$.

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

Finally, we define the set $\mathcal{C}^E$. We say that two vertices $v^i_a, v^j_b \in V(G)$ have a *conflict* if $i \neq j$ and $\{v^i_a, v^j_b\} \notin E(G)$. For every two conflicting vertices $v^i_a, v^j_b \in V(G)$, we have the constraint $C^E_{(i,a),(j,b)} = (\{x_i, \overline{x}_i, x_j, \overline{x}_j\}, \mathcal{R}^E_{(i,a),(j,b)})$ of arity 4, where $\mathcal{R}^E_{(i,a),(j,b)} = \{\{(x_i, a+1)\}, \{(x_j, b+1)\}, \{(\overline{x}_i, n'-a+2)\}, \{(\overline{x}_j, n'-b+2)\}\}$.[1] Next, we define $\mathcal{C}^E = \{C^E_{(i,a),(j,b)} : v^i_a, v^j_b \in V(G)$ 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.

3.2 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* $\texttt{red}(G, \chi, k) = (X, \mathcal{C}, n', W)$ *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 $\texttt{id}(i)$ be the integer in $[n']$ such that $v^i_{\texttt{id}(i)} \in V(C)$. Then, we define an assignment $\alpha : X \to [n']$ as follows. For all $i \in [k]$, set $\alpha(x_i) = \texttt{id}(i)$ and $\alpha(\overline{x}_i) = n' - \texttt{id}(i) + 1$.

Let us first observe that $\texttt{cost}(\alpha) = \sum_{i=1}^{k} (\alpha(x_i) + \alpha(\overline{x}_i)) = k(n'+1) = W$. Now, note that for all $i \in [k]$, the mini-constraint $R^V_{i,\texttt{id}(i)}$ is satisfied by α, and therefore R^V_i is satisfied by α. Thus, the set $\mathcal{C}^V$ is also satisfied by α. Next, consider some constraint $C^E_{(i,a),(j,b)} \in \mathcal{C}^E$. Then, we have that the two vertices $v^i_a, v^j_b \in V(G)$ have a conflict, which means that $i \neq j$ and $\{v^i_a, v^j_b\} \notin E(G)$. Since C is a multicolored clique in G of size k, we have that at least one vertex in $\{v^i_a, v^j_b\}$ does not belong to $V(C)$. Without loss of generality, suppose that this vertex is v^i_a, that is, $\texttt{id}(i) \neq a$. In this case, either $\texttt{id}(i) \geq a+1$, in which case $\alpha(x_i) \geq a+1$ and then α satisfies $(x_i, a+1)$, or $\texttt{id}(i) \leq a-1$, in which case $\alpha(\overline{x}_i) \geq n' - (a-1) + 1 = n' - a + 2$ and then α satisfies $(\overline{x}_i, n'-a+2)$. In both cases, we deduce that α satisfies $C^E_{(i,a),(j,b)}$. Since the choice of this constraint was arbitrary, we have that α satisfies $\mathcal{C}^E$. Overall, we have that α is an assignment of cost at most W that satisfies $\mathcal{C}$, and therefore $(X, \mathcal{C}, n', W)$ is a Yes-instance of 4-Monotone min-CSP. $\square$

We proceed by proving the reverse direction.

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

Lemma 3.3. *Let (G, χ, k) be an instance of* Multicolored Clique. *If* $\texttt{red}(G, \chi, k) = (X, \mathcal{C}, n', W)$ *is a* Yes*-instance of* 4-Monotone min-CSP*, then (G, χ, k) is a* Yes*-instance of* Multicolored Clique.

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

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 $v^i_{\texttt{id}(i)}, v^j_{\texttt{id}(j)} \in V(C)$ such that $\{v^i_{\texttt{id}(i)}, v^j_{\texttt{id}(j)}\} \notin E(G)$. Then, $v^i_{\texttt{id}(i)}$ and $v^j_{\texttt{id}(j)}$ have a conflict. Since α satisfies $\mathcal{C}^E$, it in particular satisfies $C^E_{(i,\texttt{id}(i)),(j,\texttt{id}(j))} = (\{x_i, \overline{x}_i, x_j, \overline{x}_j\}, \mathcal{R}^E_{(i,\texttt{id}(i)),(j,\texttt{id}(j))})$, where $\mathcal{R}^E_{(i,\texttt{id}(i)),(j,\texttt{id}(j))} = \{\{(x_i, \texttt{id}(i) + 1)\}, \{(x_j, \texttt{id}(j) + 1)\}, \{(\overline{x}_i, n' - \texttt{id}(i) + 2)\}, \{(\overline{x}_j, n' - \texttt{id}(j) + 2)\}\}$. In other words, at least one of the following four conditions is satisfied: **(i)** $\alpha(x_i) \geq \texttt{id}(i) + 1$, which contradicts that $\texttt{id}(i) = \alpha(x_i)$; **(ii)** $\alpha(x_j) \geq \texttt{id}(j) + 1$, which contradicts that $\texttt{id}(j) = \alpha(x_j)$; **(iii)** $\alpha(\overline{x}_i) \geq n' - \texttt{id}(i) + 2$, which contradicts that $n' - \texttt{id}(i) + 1 = \alpha(\overline{x}_i)$; **(iv)** $\alpha(\overline{x}_j) \geq n' - \texttt{id}(j) + 2$, which contradicts that $n' - \texttt{id}(j) + 1 = \alpha(\overline{x}_j)$. We thus conclude that C is a multicolored clique on k vertices, and therefore (G, χ, k) is a Yes-instance of Multicolored Clique. □

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 $\mathcal{O}(f(k) \cdot n^{o(k)})$ 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 $\texttt{red}(G, \chi, k) = (X, \mathcal{C}, n', W)$ of 4-Monotone min-CSP in polynomial time. Then, it calls algorithm $\mathcal{A}$ with $(X, \mathcal{C}, n', W)$ 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 $\mathcal{O}(f(k) \cdot n^{o(k)})$, which contradicts Proposition 3.1. This concludes the proof. □

4 Reduction to Graph Coloring

In this section, we prove Theorem 1 by presenting a reduction from 4-Monotone min-CSP to Graph Coloring.

4.1 Construction

Let $(X, \mathcal{C}, n, W)$ be an instance of 4-Monotone min-CSP, where $|X| = 2^k$. Here, we denote $X = \{x_0, x_1, \ldots, x_{2^k-1}\}$ (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 $(X, \mathcal{C}, n, W)$ is a Yes-instance of 4-Monotone min-CSP (to see this, simply assign n to every variable). Finally, without loss of generality, we assume that every variable belongs to at most one pair in any individual mini-constraint, as otherwise the mini-constraint contains a redundant inequality. In

what follows, we construct an instance $\texttt{red}(X, \mathcal{C}, n, W) = (G, k')$ of Graph Coloring, where $k' = 2k + \mathcal{O}(1)$ is the neighborhood-width of G. (Note that, as will be formally proved later, the parameter changes from 2^k to $\mathcal{O}(k)$).

Assignment Encoder. We first create k vertex-disjoint cliques, $B^1, B^2, \ldots B^k$, each on $2^k n$ new vertices. We denote $\mathcal{B} = \{B^1, B^2 \ldots, B^k\}$. 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^i_0 and B^i_1. In addition, we add another clique, called $B^\star$, on $2^k n - W$ new vertices, and denote $\mathcal{B}^\star = \mathcal{B} \cup \{B^\star\}$. 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 $B^\star$. An illustration of the construction up to this point is given in Fig. 1.

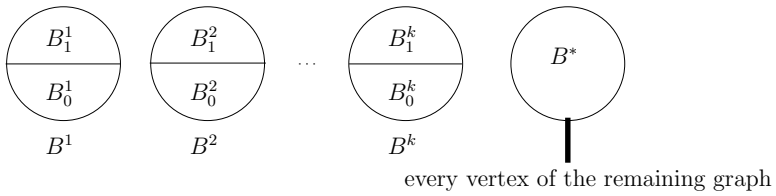

Figure 1: Assignment Encoder. The thick line is used to denote all edges joining B^* with the remaining vertices of the graph.

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^b_0 and B^b_1 indicate whether that bit is 0 or 1, respectively. Moreover, for all $i \in [2^k - 1]_0$ and $b \in [k]$, let $\texttt{bit}(i, b)$ denote the b^{st} bit of the ID i. That is, $i = \sum_{b=1}^{k} \texttt{bit}(i, b) 2^{b-1}$. 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 $\mathcal{B}[i] = \{B^b_{\texttt{bit}(i,b)} : b \in [k]\}$, and also let us denote the complementary set by $\overline{\mathcal{B}}[i] = \{B^b_{1-\texttt{bit}(i,b)} : b \in [k]\}$.

We will later ensure (as will be clear in the proof) that all of the cliques in $\mathcal{B}^\star$ 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 $B^\star$ 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 $\mathcal{B}^\star$. 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 colors 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 $B^\star$. 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 $B^\star$. 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. Let $\mathcal{R}[i]$ denote that set of mini-constraints in $\mathcal{R}$ that involve x_i. We arbitrarily partition $A^{(C,i)}$ into $|\mathcal{R}[i]|+1$ vertex-disjoint cliques, denoted by $A_R^{(C,i)}$ for all $R \in \mathcal{R}[i]$ and $A_\star^{(C,i)}$, where for all $R \in \mathcal{R}[i]$, the clique $A_R^{(C,i)}$ contains n vertices, and the clique $A_\star^{(C,i)}$ contains $n(M-|\mathcal{R}[i]|)$ 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 Fig. 2.

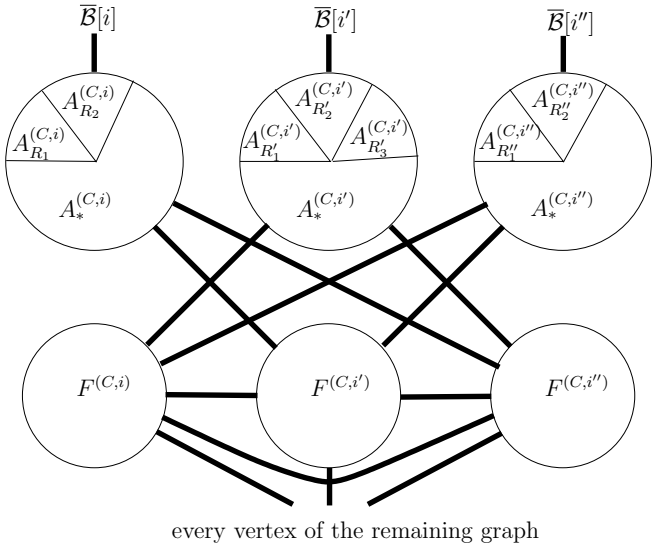

Figure 2: Constraint Variable; $C = (X', \mathcal{R})$, $X' = \{i, i', i''\}$, $\mathcal{R}[i] = \{R_1, R_2\}$, $\mathcal{R}[i'] = \{R'_1, R'_2, R'_3\}$ and $\mathcal{R}[i''] = \{R''_1, R''_2\}$. The thick lines are used to denote all edges joining the cliques $A^{(C,i)}$, $A^{(C,i')}$, $A^{(C,i'')}$, $F^{(C,i)}$, $F^{(C,i')}$ and $F^{(C,i'')}$ with each other and the remaining vertices of the graphs.

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 $\mathcal{R}[i]$, then exactly $n-c$ colors of the third type will be available, which would mean that 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 $B^\star$ 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 refine the independent set $I^{(C,R)}$. For every $i \in [2^k - 1]_0$ such that $R \in \mathcal{R}[i]$, let $I_i^{(C,R)}$ denote a subset of $I^{(C,R)}$ of size $(n-c)$ where c is the unique integer in $[n]$ satisfying $(x_i, c) \in R$, so that for all distinct $i, j \in [2^k - 1]_0$ such that $R \in \mathcal{R}[i] \cap \mathcal{R}[j]$, 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 Fig. 3.

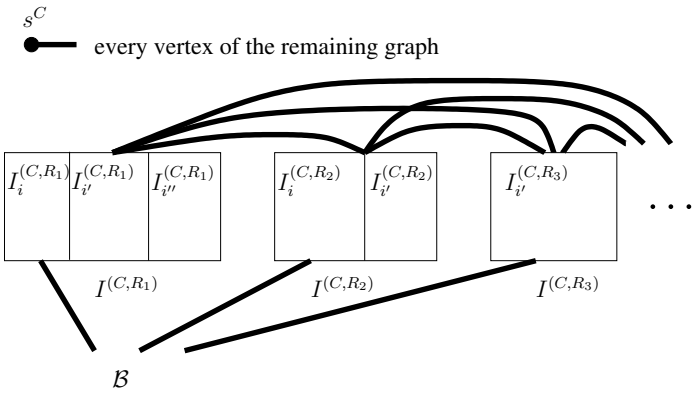

Figure 3: Mini-Constraint Selector. The thick lines are used to denote all edges joining s^C and the independent sets $I^{(C,R_1)}, I^{(C,R_2)}, I^{(C,R_3)}, \ldots$ with each other and the remaining vertices of the graphs.

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 necessary 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 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 $\widehat{A}_R^{(C,i)}$ 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 $G[I_i^{(C,R)} \cup V(\widehat{A}_R^{(C,i)})]$, where each new edge has one endpoint in $I_i^{(C,R)}$ and the other endpoint in $\widehat{A}_R^{(C,i)}$. 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 Fig. 4.

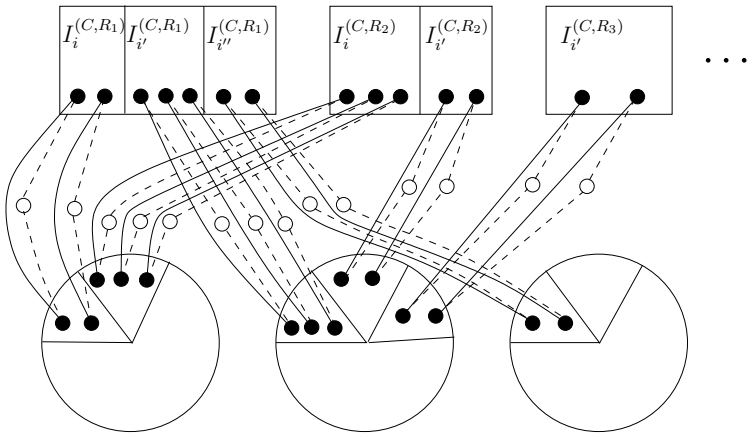

Figure 4: Matching Edges and Vertices. The white bullets are used to depict the vertices v_e and the incident dashed lines show non-edges.

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 $I^{(C,R)}$ 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 $\widehat{A}_R^{(C,i)}$, 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 $(n-c)$ 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. Denote $\eta = 2^k n + ((n-1)M \sum_{(X',\mathcal{R})\in\mathcal{C}} |X'|) + |\mathcal{C}| + (\sum_{(X',\mathcal{R})\in\mathcal{C}} \sum_{R\in\mathcal{R}} \sum_{(x_i,c)\in R} (n - c))$. (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.

4.2 Correctness

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

Lemma 4.1. *Let $(X, \mathcal{C}, n, W)$ be an instance of* 4-Monotone min-CSP *with $|X| = 2^k$. If $(X, \mathcal{C}, n, W)$ is a* Yes*-instance of* 4-Monotone min-CSP*, then the chromatic number of G in* $\texttt{red}(X, \mathcal{C}, n, W) = (G, k')$ *is at most η.*

Proof. Suppose that $(X, \mathcal{C}, n, W)$ is a Yes-instance of 4-Monotone min-CSP, and let α be an assignment of cost at most W that satisfies $\mathcal{C}$. Without loss of generality, we can assume that $\texttt{cost}(\alpha) = W$, we otherwise we can increase the value assigned to some variables so that this condition will be satisfied. In what follows, we construct a proper coloring $\chi : V(G) \to [\eta]$. This would imply that the chromatic number of G is at most η, which would conclude the proof. First,

we use $\mu = ((n-1)M \cdot \sum_{(X',\mathcal{R})\in\mathcal{C}} |X'|) + |\mathcal{C}| + (\sum_{(X',\mathcal{R})\in\mathcal{C}} \sum_{R\in\mathcal{R}} \sum_{(x_i,c)\in R}(n-c))$ 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 $\texttt{alloc}(i, B)$ be some (arbitrarily chosen) set of $\alpha(x_i)$ vertices in B, so that for all distinct $i, j \in [k]$ and $B \in \mathcal{B}[i] \cap \mathcal{B}[j]$, $\texttt{alloc}(i, B) \cap \texttt{alloc}(j, B) = \emptyset$. 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 $\texttt{alloc}$ can be well-defined. Moreover, for all $i \in [2^k - 1]_0$, let $\texttt{col}(i)$ denote some (arbitrarily chosen) set of $\alpha(x_i)$ colors in $[\mu + W] \setminus [\mu]$, so that for all distinct $i, j \in [k]$, $\texttt{col}(i) \cap \texttt{col}(j) = \emptyset$. Since $\texttt{cost}(\alpha) \leq W$, there is a sufficient number of colors in $[\mu + W] \setminus [\mu]$ to ensure that $\texttt{col}$ 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 $\texttt{alloc}(i, B)$ with distinct colors from $\texttt{col}(i)$. Clearly, all the vertices of the same clique in $\mathcal{B}$ that we have colored so far received distinct colors (because for all distinct $i, j \in [k]$, $\texttt{col}(i) \cap \texttt{col}(j) = \emptyset$), 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 $\texttt{cost}(\alpha) = W$). 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 $|V(B^\star)| = 2^k n - W$ and there are no edges between vertices in $B^\star$ and vertices in the cliques in $\mathcal{B}$, we can also color every vertex in $B^\star$ 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 $\mathcal{I}^C$. 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 $\mathcal{I}^C$ and a vertex in $A^{(\widehat{C},j)}$ for some $x_j \in \widehat{X}$ or in $\mathcal{I}^{\widehat{C}}$, 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 $\mathcal{C}$. For all $i \in [2^k - 1]_0$ such that $x_i \in X'$, let c_i denote the (unique) integer in $[n]$ satisfying $(x_i, c_i) \in R$.

Let us begin by coloring the vertices in the independent sets in $\mathcal{I}^C$. 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 $\mathcal{I}^C \setminus \{I^{(C,R)}\}$ 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 $\texttt{col}(i) \geq c_i$. Note that all the vertices colored by c_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 $\texttt{col}(i)$. 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. □

Towards 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 4.1. *Let $(X, \mathcal{C}, n, W)$ be an instance of* 4-Monotone min-CSP *with $|X| = 2^k$, and denote $\texttt{red}(X, \mathcal{C}, n, W) = (G, k')$. Let χ be a proper coloring of G. Then, the assignment $\alpha_\chi : X \to [n]_0$ is defined as follows. For all $i \in [2^k - 1]_0$, denote $\texttt{col}_\chi(i) = \{j \in \mathsf{ima}(\chi) :$ every clique in $\mathcal{B}[i]$ has a vertex colored j by $\chi\} \setminus \chi(V(B^\star))$. Then, for all $x_i \in X$, $\alpha(x_i) \triangleq |\texttt{col}_\chi(i)|$.*

It will also be convenient for us to use the following notation: In the context of an output $\texttt{red}(X, \mathcal{C}, n, W) = (G, k')$, 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 4.1. *Let $(X, \mathcal{C}, n, W)$ be an instance of* 4-Monotone min-CSP *with $|X| = 2^k$, and denote $\texttt{red}(X, \mathcal{C}, n, W) = (G, k')$. 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. □

By using Observation 4.1, we derive the following result.

Lemma 4.2. *Let $(X, \mathcal{C}, n, W)$ be an instance of* 4-Monotone min-CSP *with $|X| = 2^k$, and denote $\texttt{red}(X, \mathcal{C}, n, W) = (G, k')$. Let χ be a proper coloring of G such that $\mathsf{ima}(\chi) \subseteq [\eta]$. For all $i \in [k]$, $\chi(V(B^i)) = [\eta] \setminus \chi(D)$ and $\chi(V(B^\star)) \subseteq \chi(V(B^i))$.*

Proof. First, notice that $|D| = \eta - 2^k n$ and that for all $i \in [k]$, $|B^i| = 2^k n$. By Observation 4.1, for all $i \in [k]$, we have that $\chi(B^i) \subseteq \mathsf{ima}(\chi) \setminus \chi(D)$. However, since $\mathsf{ima}(\chi) \subseteq [\eta]$, this implies that for all $i \in [k]$, indeed $\chi(B^i) = \mathsf{ima}(\chi) \setminus \chi(D)$. Since every vertex in $B^\star$ 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 $\chi(V(B^\star)) \subseteq \chi(V(B^i))$. □

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

Lemma 4.3. *Let $(X, \mathcal{C}, n, W)$ be an instance of* 4-Monotone min-CSP *with $|X| = 2^k$, and denote $\texttt{red}(X, \mathcal{C}, n, W) = (G, k')$. Let χ be a proper coloring of G such that $\mathsf{ima}(\chi) \subseteq [\eta]$. Then, $\texttt{cost}(\alpha_\chi) \leq W$.*

Proof. By the definition of α_χ (Definition 4.1), to prove that the lemma is correct, we need to show that $\sum_{i=0}^{2^k-1} |\texttt{col}_\chi(i)| \leq W$. 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 $\texttt{col}(\cdot)$ we have that $\texttt{col}(i) \cap \texttt{col}(j) = \emptyset$. Thus, $\sum_{i=0}^{2^k-1} |\texttt{col}_\chi(i)| = |\bigcup_{i=0}^{2^k-1} \texttt{col}(i)|$. Now, note that for any $i \in [k]$,

$\bigcup_{i=0}^{2^k-1} \texttt{col}(i) \subseteq \chi(V(B^i)) \setminus \chi(V(B^\star))$, $|V(B^i)| = 2^k n$ and $|V(B^\star)| = 2^k n - W$. Moreover, by Lemma 4.2, for any $i \in [k]$, $\chi(V(B^\star)) \subseteq \chi(V(B^i))$. Therefore, for any $i \in [k]$, $|\bigcup_{i'=0}^{2^k-1} \texttt{col}(i')| = |V(B^i)| - |V(B^\star)| = W$. We thus have that $\texttt{cost}(\alpha_\chi) \leq W$. □

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

Definition 4.2. *Let $(X, \mathcal{C}, n, W)$ be an instance of* 4-Monotone min-CSP *with $|X| = 2^k$, and denote $\texttt{red}(X, \mathcal{C}, n, W) = (G, k')$. A function χ is a* nice coloring *of G if it is a proper coloring of G, $\mathsf{ima}(\chi) \subseteq [\eta]$, and for all $C = (X', \mathcal{R}) \in \mathcal{C}$, the two following conditions hold.*

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

Lemma 4.4. *Let $(X, \mathcal{C}, n, W)$ be an instance of* 4-Monotone min-CSP *with $|X| = 2^k$, and denote $\texttt{red}(X, \mathcal{C}, n, W) = (G, k')$. If there exists a proper coloring χ of G such that $\mathsf{ima}(\chi) \subseteq [\eta]$, then there also exists a nice coloring $\widehat{\chi}$ of G.*

Proof. Let χ be a proper coloring of G such that $\mathsf{ima}(\chi) \subseteq [\eta]$. Consider some constraint $C = (X', \mathcal{R}) \in \mathcal{C}$. 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^C_χ 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 $I^{(C,R^C_\chi)}$ 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 $R \in \mathcal{R} \setminus \{R^C_\chi\}$ 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 $\bigcup_{x_i \in X'} M^{(C,i)}_R$ incident to v. By Lemma 4.2 and since $\mathsf{ima}(\widehat{\chi}) \subseteq [\eta]$, 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 $\mathcal{C}$ individually. This completes the proof. □

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 $\texttt{red}(X, \mathcal{C}, n, W) = (G, k')$ and a nice coloring χ of G, for all $C = (X', \mathcal{R}) \in \mathcal{C}$ and $x_i \in X'$, let $c^{(C,i)}_\chi$ denote unique integer in $[n]$ satisfying $(x_i, c^{(C,i)}_\chi) \in R^C_\chi$.

Lemma 4.5. *Let $(X, \mathcal{C}, n, W)$ be an instance of* 4-Monotone min-CSP *with $|X| = 2^k$, and denote $\texttt{red}(X, \mathcal{C}, n, W) = (G, k')$. Let χ be a nice coloring of G. For all $\widehat{C} = (\widehat{X}, \widehat{\mathcal{R}}) \in \mathcal{C}$ and $x_{\widehat{i}} \in \widehat{X}$, at most $n - c^{(\widehat{C},\widehat{i})}_\chi$ vertices in $A^{(\widehat{C},\widehat{i})}$ are colored as vertices in $\bigcup_{C=(X',\mathcal{R}) \in \mathcal{C}} \bigcup_{x_i \in X'} \widehat{M}^{(C,i)}_R$.*

Proof. Let us fix some $\widehat{C} = (\widehat{X}, \widehat{\mathcal{R}}) \in \mathcal{C}$ and $x_{\widehat{i}} \in \widehat{X}$. First, note that all the vertices in $A^{(\widehat{C},\widehat{i})}$ are adjacent to all the vertices in $\bigcup_{C=(X',\mathcal{R}) \in \mathcal{C} \setminus \{\widehat{C}\}} \bigcup_{x_i \in X'} \widehat{M}^{(C,i)}_R$, and therefore they can clearly not be colored as the vertices in this set. Moreover, for all $R \in \widehat{\mathcal{R}} \setminus \{R^{\widehat{C}}_\chi\}$ and $v \in I^{(\widehat{C},R)}$, we have that v is colored as v^e where e is the edge in $\bigcup_{x_i \in X'} M^{(C,i)}_R$ 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 $\widehat{M}_{R_\chi^{\widehat{C}}}^{(\widehat{C},\widehat{i})}$. Since $|\widehat{M}_{R_\chi^{\widehat{C}}}^{(\widehat{C},\widehat{i})}| = n - c_\chi^{(\widehat{C},\widehat{i})}$, this completes the proof. □

Lemma 4.6. *Let $(X,\mathcal{C},n,W)$ be an instance of* 4-Monotone min-CSP *with $|X| = 2^k$, and denote $\mathtt{red}(X,\mathcal{C},n,W) = (G,k')$. 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 $B^\star$ is colored j.*

Proof. Fix some $C = (X',\mathcal{R}) \in \mathcal{C}$ and $x_i \in X'$. By Observation 4.1 and since $\mathsf{ima}(\chi) \subseteq [\eta]$, 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 4.5 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 $B^\star$, the correctness of the lemma follows. □

Finally, we are ready to prove the reverse direction.

Lemma 4.7. *Let $(X,\mathcal{C},n,W)$ be an instance of* 4-Monotone min-CSP *with $|X| = 2^k$. If the chromatic number of G in $\mathtt{red}(X,\mathcal{C},n,W) = (G,k')$ is at most η, then $(X,\mathcal{C},n,W)$ is a* Yes*-instance of* 4-Monotone min-CSP*.*

Proof. Suppose that the chromatic number of G is at most η. By Lemma 4.4, G has a nice coloring χ. Let α denote the assignment $\alpha_\chi : X \to [n]_0$. By Lemma 4.3, we have that $\mathtt{cost}(\alpha) \leq W$. Now, let us consider some $C = (X',\mathcal{R}) \in \mathcal{C}$. Then, Lemma 4.6 directly implies that for all $x_i \in X'$, $\mathtt{col}(i) \geq c_\chi^{(C,i)}$. 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 $\mathcal{C}$. Hence, $(X,\mathcal{C},n,W)$ is a Yes-instance of 4-Monotone min-CSP. □

4.3 Cliquewidth

Towards the proof of Theorem 1, it remains to bound the cliquewidth of the output graph.

Lemma 4.8. *Let $(X,\mathcal{C},n,W)$ be an instance of* 4-Monotone min-CSP *with $|X| = 2^k$. The neighborhood-width of G in $\mathtt{red}(X,\mathcal{C},n,W) = (G,k')$ is at most $2k + \mathcal{O}(1)$.*

Proof. We define an ordering σ on $V(G) = \{v_1, v_2, \ldots, v_{n'}\}$ as follows. We let the first $2^k n(k+1) - W$ vertices in this order be all the vertices in the cliques in $\mathcal{B}^\star$, where the internal order among them is arbitrary. Notice that $B^\star$ 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 $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$, and for all $j \in [m]$, denote $C_j = (X_j, \mathcal{R}_j)$. For $j = 1, 2, \ldots, m$, 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 $\mathcal{R}_j = \{R_1, R_2, \ldots, R_r\}$. For $p = 1,2,\ldots,r$ (outer loop) and $q = 1,2,\ldots,|X_j|$ (inner loop), we will consecutively insert all the vertices in $D^j_{p,q} \triangleq V(\widehat{A}_{R_p}^{(C_j,q)}) \cup I_q^{(C_j,R_p)} \cup \widehat{M}_{R_p}^{(C_j,q)}$ 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 $D^j_{p',q'}$ where $(p',q') < (p,q)$, that is, either $1 \leq p' < p$ or both $p' = p$ and $1 \leq q' < q$. Now, note that $\bigcup_{(p',q')<(p,q)} D^j_{(p',q')}$ consists of seven modules with respect to $V(G) \setminus (\bigcup_{j'=0}^{j-1} D^{j'}) \cup (\bigcup_{(p',q')<(p,q)} D^j_{(p',q')})$—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 $M_{R_p}^{(C_j,q)} = \{e_1, e_2, \ldots, e_s\}$. For $\ell = 1,2,\ldots,s$, 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 $D^j_{p,q}$) 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 $\mathtt{nw}(G,\sigma) \leq 2k+24$. Therefore, $\mathtt{nw}(G) \leq 2k + \mathcal{O}(1)$. This completes the proof. □

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

Theorem 2. *Unless ETH fails,* Graph Coloring *cannot be solved in time $\mathcal{O}(f(k) \cdot n^{2^{o(k)}})$ 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 $\mathcal{O}(f(k) \cdot n^{2^{o(k)}})$ 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 $\mathcal{O}(f(k') \cdot n'^{2^{o(k')}})$. Now, consider the following algorithm $\mathcal{B}$ for 4-Monotone min-CSP. Given an instance $(X, \mathcal{C}, n, W)$ of 4-Monotone min-CSP where $k = 2^{\widehat{k}}$, algorithm $\mathcal{B}$ first constructs the instance $\mathtt{red}(X, \mathcal{C}, n, W) = (G, k')$ 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 4.1 and 4.7, algorithm $\mathcal{B}$ is correct. Furthermore, in the output instance $n' = n^{\mathcal{O}(1)}$, and by Lemma 4.8, we also have that $k' = 2\widehat{k} + \mathcal{O}(1) = 2\log_2 k + \mathcal{O}(1)$. Thus, algorithm $\mathcal{B}$ solves 4-Monotone min-CSP in time

$\mathcal{O}(f(k') \cdot n'^{2^{o(k')}}) = \mathcal{O}(f(\log k) \cdot (n^{\mathcal{O}(1)})^{2^{o(\log k)}}) = \mathcal{O}(g(k)n^{o(k)})$ for some function g of k, which contradicts Lemma 3.1. This concludes the proof. □

By Proposition 2.1, we have the following corollary to Theorem 2.

Corollary 4.1. *Unless ETH fails,* Graph Coloring *cannot be solved in time* $\mathcal{O}(f(k) \cdot n^{2^{o(k)}})$ *for any function* f *of* k*, where* k *is the linear cliquewidth of* G*.*

Finally, due to Observation 2.1, Theorem 1 follows as a consequence of Corollary 4.1.

5 Conclusion

In this paper we proved that unless the ETH fails, GC can not be solved in time $\mathcal{O}(f(k) \cdot n^{2^{o(k)}})$. where k is the (linear) cliquewidth of the input graph. At this point, the complexity of MC, EDS, GC and HP on graphs of bounded cliquewidth is quite well-understood. On the other hand, pinning down the right exponent of n for these problems on graphs of *rankwidth* k remains open. The more intriguing open problem remains the complexity of computing the cliquewidth of a graph. To the best of our knowledge, it is consistent with current knowledge that determining whether G has cliquewidth k is FPT or that that determining whether G has cliquewidth 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 cliquewidth k is NP-complete for some fixed constant $k \geq 5$.

Acknowledgement. We thank Michał Pilipczuk for insightful discussions.

References

[1] S. Arnborg, J. Lagergren, and D. Seese, *Easy problems for tree-decomposable graphs*, J. Algorithms, 12 (1991), pp. 308–340. 1, 2

[2] B. Bergougnoux, M. M. Kanté, and O. Kwon, *An optimal XP algorithm for hamiltonian cycle on graphs of bounded clique-width*, to appear in WADS 2017, Also see: abs/1702.06095 (2017). 2

[3] U. Bertelé and F. Brioschi, *Nonserial dynamic programming*, Academic Press, 1972. 1

[4] H. L. Bodlaender, *A partial k-arboretum of graphs with bounded treewidth*, Theoret. Comput. Sci., 209 (1998), pp. 1–45. 1

[5] R. B. Borie, R. G. Parker, and C. A. Tovey, *Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families*, Algorithmica, 7 (1992), pp. 555–581. 2

[6] B. Bui-Xuan, J. A. Telle, and M. Vatshelle, *Boolean-width of graphs*, Theor. Comput. Sci., 412 (2011), pp. 5187–5204. 1

[7] C. Calabro, R. Impagliazzo, and R. Paturi, *The complexity of satisfiability of small depth circuits*, in Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers, 2009, pp. 75–85. 5

[8] J. Chen, X. Huang, I. A. Kanj, and G. Xia, *Strong computational lower bounds via parameterized complexity*, J. Comput. Syst. Sci., 72 (2006), pp. 1346–1367. 3

[9] D. G. Corneil, M. Habib, J.-M. Lanlignel, B. A. Reed, and U. Rotics, *Polynomial time recognition of clique-width ≤ 3 graphs (extended abstract)*, in LATIN'00, vol. 1776 of LNCS, Springer, 2000, pp. 126–134. 1

[10] D. G. Corneil and U. Rotics, *On the relationship between clique-width and treewidth*, SIAM J. Comput., 34 (2005), pp. 825–847. 1

[11] B. Courcelle, *The monadic second-order logic of graphs. III. Tree-decompositions, minors and complexity issues*, RAIRO Inform. Théor. Appl., 26 (1992), pp. 257–286. 1, 2

[12] B. Courcelle, J. A. Makowsky, and U. Rotics, *Linear time solvable optimization problems on graphs of bounded clique-width*, Theory Comput. Syst., 33 (2000), pp. 125–150. 1

[13] B. Courcelle and S. Olariu, *Upper bounds to the clique width of graphs*, Discrete Appl. Math., 101 (2000), pp. 77–114. 1

[14] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized Algorithms*, Springer, 2015. 3, 5

[15] M. Cygan, M. Pilipczuk, and M. Pilipczuk, *Known algorithms for edge clique cover are probably optimal*, SIAM J. Comput., 45 (2016), pp. 67–83. 2

[16] R. Diestel, *Graph Theory*, Springer-Verlag, Heidelberg, 4th ed., 2010. 3

[17] R. G. Downey and M. R. Fellows, *Fundamentals of Parameterized Complexity*, Texts in Computer Science, Springer, 2013. 5

[18] W. Espelage, F. Gurski, and E. Wanke, *How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time*, in WG'01, vol. 2204 of LNCS, Springer, 2001, pp. 117–128. 2

[19] M. R. Fellows, F. A. Rosamond, U. Rotics, and S. Szeider, *Clique-width is NP-complete*, SIAM J. Discrete Math., 23 (2009), pp. 909–939. 1

[20] F. V. Fomin, P. A. Golovach, D. Lokshtanov, and S. Saurabh, *Algorithmic lower bounds for problems parameterized by clique-width*, in SODA, 2010, pp. 493–502. 2

[21] ——, *Almost optimal lower bounds for problems parameterized by clique-width*, SIAM J. Comput., 43 (2014), pp. 1541–1563. 2

[22] M. Frick and M. Grohe, *The complexity of first-order and monadic second-order logic revisited*, Ann. Pure Appl. Logic, 130 (2004), pp. 3–31. 2

[23] M. U. Gerber and D. Kobler, *Algorithms for vertex-partitioning problems on graphs with fixed clique-width*, Theoret. Comput. Sci., 299 (2003), pp. 719–734. 2

[24] O. Giménez, P. Hliněný, and M. Noy, *Computing the tutte polynomial on graphs of bounded clique-width*, SIAM J. Discret. Math., 20 (2006). 2

[25] B. Godlin, T. Kotek, and J. A. Makowsky, *Evaluations of graph polynomials*, in WG, vol. 5344 of Lecture Notes in Computer Science, 2008, pp. 183–194. 2

[26] F. Gurski, *Linear layouts measuring neighbourhoods in graphs*, Discrete Mathematics, 306 (2006), pp. 1637–1650. 2, 4

[27] R. Halin, *S-functions for graphs*, Journal of geometry, 8 (1976), pp. 171–186. 1

[28] P. Hliněný and S. il Oum, *Finding branch-decompositions and rank-decompositions*, SIAM J. Comput., 38 (2008), pp. 1012–1032. 1

[29] P. Hliněný, S.-il Oum, D. Seese, and G. Gottlob, *Width parameters beyond tree-width and their applications*, Comput. J., 51 (2008), pp. 326–362. 1

[30] R. Impagliazzo and R. Paturi, *On the complexity of k-SAT*, J. Comput. Syst. Sci., 62 (2001), pp. 367–375. 5

[31] R. Impagliazzo, R. Paturi, and F. Zane, *Which problems have strongly exponential complexity?*, J. Comput. Syst. Sci., 63 (2001), pp. 512–530. 5

[32] D. Kobler and U. Rotics, *Polynomial algorithms for partitioning problems on graphs with fixed clique-width (extended abstract)*, in SODA'01, ACM-SIAM, 2001, pp. 468–476. 2

[33] ——, *Edge dominating set and colorings on graphs with fixed clique-width*, Discrete Appl. Math., 126 (2003), pp. 197–221. 2

[34] M. Lampis, *Model checking lower bounds for simple graphs*, Logical Methods in Computer Science, 10 (2014). 2

[35] D. Lokshtanov, D. Marx, and S. Saurabh, *Lower bounds based on the exponential time hypothesis*, Bulletin of the EATCS, 105 (2011), pp. 41–72. 5

[36] J. Makowsky, U. Rotics, I. Averbouch, and B. Godlin, *Computing graph polynomials on graphs of bounded clique-width*, in WG'06, LNCS, Springer, 2006, pp. 191–204. 2

[37] D. Marx and V. Mitsou, *Double-exponential and triple-exponential bounds for choosability problems parameterized by treewidth*, in 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy, vol. 55 of LIPIcs, 2016, pp. 28:1–28:15. 2

[38] S.-il, *Approximating rank-width and clique-width quickly*, ACM Trans. Algorithms, (2008), p. to appear. 1

[39] S.-il Oum and P. Seymour, *Approximating clique-width and branch-width*, J. Combin. Theory Ser. B, 96 (2006), pp. 514–528. 1

[40] M. Rao, *MSOL partitioning problems on graphs of bounded treewidth and clique-width*, Theoret. Comput. Sci., 377 (2007), pp. 260–267. 2

[41] N. Robertson and P. D. Seymour, *Graph minors. III. planar tree-width*, J. Comb. Theory, Ser. B, 36 (1984), pp. 49–64. 1

[42] K. Suchan and I. Todinca, *On powers of graphs of bounded NLC-width (clique-width)*, Discrete Appl. Math., 155 (2007), pp. 1885–1893. 2

[43] E. Wanke, *k-NLC graphs and polynomial algorithms*, Discrete Appl. Math., 54 (1994), pp. 251–266. 1, 2