XNLP-Completeness for Parameterized Problems on Graphs with a Linear Structure

Hans L. Bodlaender
Utrecht University, The Netherlands

Carla Groenland
Utrecht University, The Netherlands

Hugo Jacob
ENS Paris-Saclay, France

Lars Jaffke
University of Bergen, Norway

Paloma T. Lima
IT University of Copenhagen, Denmark

Abstract

In this paper, we showcase the class XNLP as a natural place for many hard problems parameterized by linear width measures. This strengthens existing $W[1]$-hardness proofs for these problems, since XNLP-hardness implies $W[t]$-hardness for all t. It also indicates, via a conjecture by Pilipczuk and Wrochna [ToCT 2018], that any XP algorithm for such problems is likely to require XP space.

In particular, we show XNLP-completeness for natural problems parameterized by pathwidth, linear clique-width, and linear mim-width. The problems we consider are Independent Set, Dominating Set, Odd Cycle Transversal, (q-)Coloring, Max Cut, Maximum Regular Induced Subgraph, Feedback Vertex Set, Capacitated (Red-Blue) Dominating Set, and Bipartite Bandwidth.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases parameterized complexity, XNLP, linear clique-width, W-hierarchy, pathwidth, linear mim-width, bandwidth

Digital Object Identifier 10.4230/LIPIcs.IPEC.2022.8

Related Version *Full Version*: https://arxiv.org/abs/2201.13119

Funding *Carla Groenland*: Supported by the European Union's Horizon 2020 research and innovation programme under the ERC grant CRACKNP (number 853234) and the Marie Skłodowska-Curie grant GRAPHCOSY (number 101063180).
Lars Jaffke: Supported by the Norwegian Research Council (project number 274526) and the Meltzer Research Fund.

1 Introduction

Since the inception of parameterized complexity in the late 1980s and early 1990s, much research has been done on establishing the complexity of parameterized problems. Typically one is particularly interested in either designing FPT-algorithms for these problems, or to prove them $W[t]$-hard, for some t, which provides evidence that such a problem is not likely to be fixed-parameter tractable. As opposed to the classical P versus NP-complete setting, the question of membership in some class of the W-hierarchy is often much less clear. While some natural problems such as Independent Set and Dominating Set are known to be $W[1]$-complete and $W[2]$-complete, respectively, many other problems are unknown to

© Hans L. Bodlaender, Carla Groenland, Hugo Jacob, Lars Jaffke, and Paloma T. Lima;
licensed under Creative Commons License CC-BY 4.0

be complete for a class of parameterized problems, and even conjectured not to be in the W-hierarchy. Recently, building upon work by Elberfeld et al. [12], Bodlaender et al. [4] introduced a complexity class called XNLP, which gives a way of addressing this question.

The class XNLP consists of the parameterized problems that can be solved with a nondeterministic algorithm that uses $f(k)\log n$ space and $f(k)n^c$ time, where f is a computable function, n is the input size, k is the parameter and c is a constant. In particular, XNLP-hardness implies $W[t]$-hardness for all t. Therefore it is unlikely that any XNLP-hard problem is complete for some $W[t]$.

One success story within parameterized algorithms and complexity is the use of width measures of graphs as parameters (see, e.g., [9]). Typically, such width measures are defined in terms of a tree-like decomposition of a graph, and the width describes the complexity of the decomposition, and therefore, in turn, of the graph. Such width measures also have linear variants, where the decomposition resembles a path instead of a tree. In this work, we provide evidence that the class *XNLP is the "natural home" for hard problems parameterized by linear width measures.*

Let us give some intuitive explanation why this is the case. A typical dynamic programming algorithm that uses such a linear decomposition stores, at each node of the path, some partial solutions associated with it. The table entries associated with the nodes are then filled in the order in which they appear on the path. If one turns such an algorithm into a nondeterministic algorithm, it often suffices at the i-th node to nondeterministically determine the table index corresponding to the correct partial solution (if it exists) from the table entry that was previously determined for the $(i-1)$-th node. In such a case, membership in XNLP follows if each single table entry of such a DP algorithm can be represented by $f(k)\log n$ bits (where k is the width) and if the nondeterministic step does not require a computation that uses significantly more space. This is often the case. Now, such an approach fails for tree-like decompositions, since even a nondeterministic algorithm might have to keep too many table entries at some point during the computation. One common situation in which this occurs is when the algorithm needs to store one table entry for each level of the decomposition. This incurs a multiplicative factor in the memory usage that depends on the height of the tree, which can be prohibitively large.

In this direction, Bodlaender et al. [4] showed that List Colouring parameterized by the pathwidth of the input graph, and Bandwidth are XNLP-complete. In this paper, we show XNLP-completeness of fundamental graph problems parameterized by linear variants of well-established width measures, such as pathwidth, linear clique-width and linear mim-width, as well as some of their logarithmic analogues.

Besides showing $W[t]$-hardness for all t, XNLP-hardness also provides insight into the space complexity of parameterized problems. Pilipczuk and Wrochna [23] proposed the following conjecture.[1]

▶ **Conjecture 1** (Slice-wise Polynomial Space Conjecture [23]). *XNLP-hard problems do not have an algorithm, that runs in $n^{f(k)}$ time and $f(k)n^c$ space, with f a computable function, k the parameter, n the input size, and c a constant.*

Typically, membership in XP for the problems studied in our paper follows from a dynamic programming approach that uses a significant amount of memory. XNLP-hardness indicates (via Conjecture 1) that dynamic programming is in some sense "optimal" (no XP algorithm can use "significantly less" memory).

---

[1] The statement of the conjecture here is equivalent to the conjecture on time and memory use for the Longest Common Subsequence problem from [23]; the name of the conjecture is taken as analogue to the naming of XP as problems that use *slice-wise polynomial time* (see [9, Section 1.1]).

Linear width measures and logarithmic analogues. The width measures we consider in this work include linear variants of arguably the most prominent measures, and some of their generalizations. Pathwidth is a linear variant of the classic treewidth parameter, which, informally speaking, measures how close a connected graph is to being a tree. In this vein, pathwidth measures how close a connected graph is to being a path. Clique-width (or, equivalently, rank-width) generalizes treewidth to several simply structured dense graphs, and its linear counterpart is called linear clique-width (linear rank-width). Maximum induced matching width [24], or mim-width for short, in turn generalizes clique-width and remains bounded even on well-studied graph classes such as interval and permutation graphs, where the clique-width is known to be unbounded. In fact, for most of these classes the *linear* mim-width is bounded.

We also introduce a new parameter that we call logarithmic linear clique-width, analogous to the parameter logarithmic pathwidth that was introduced by Bodlaender et al. [4]. For an n-vertex graph of linear clique-width k, logarithmic linear clique-width takes the value $\lceil k/\log n\rceil$. We stress the fact that XNLP-hardness parameterized by a logarithmic parameter implies that there is no algorithm solving the problem in time $2^{O(k)}n^{O(1)}$ and space $k^{O(1)}n^{O(1)}$, where k is the original parameter[2], under Conjecture 1. Such results can complement existing (S)ETH lower bounds for single exponential FPT algorithms with lower bounds on the space requirements of such algorithms.

Bipartite bandwidth. Finally, we consider a bipartite variant of the notoriously difficult [2] problem of computing the bandwidth of a graph. Here, for a bipartite graph with vertex bipartition (A, B) and bandwidth target value w, we seek an ordering α of A and an ordering β of B, such that for each edge ab, $|\alpha(a)-\beta(b)| \leq w$. We consider this problem parameterized by w, and show that it is XNLP-complete, even when the input graph is a tree.

Our results. We summarize our results in the following theorem.

▶ **Theorem 2.** *The following problems are XNLP-complete.*

(i) CAPACITATED RED-BLUE DOMINATING SET *and* CAPACITATED DOMINATING SET *parameterized by pathwidth.*

(ii) *COLORING, MAXIMUM REGULAR INDUCED SUBGRAPH, and* MAX CUT *parameterized by linear clique-width.*

(iii) q-COLORING *and* ODD CYCLE TRANSVERSAL *parameterized by logarithmic pathwidth or logarithmic linear clique-width.*

(iv) *INDEPENDENT SET, DOMINATING SET, FEEDBACK VERTEX SET, and q-COLORING for fixed $q \geq 5$ parameterized by linear mim-width.*

(v) BIPARTITE BANDWIDTH*, even if the input graph is a tree.*

Furthermore, FEEDBACK VERTEX SET parameterized by logarithmic pathwidth or logarithmic linear clique-width is XNLP-hard.

Note that Theorem 2(ii) and (iv) include the first XNLP-completeness results for graph problems with the linear clique-width and linear mim-width as parameter.

[2] Indeed, replacing k with $k' \log n$, this gives running time $2^{O(k' \log n)} = n^{O(k')}$ and space $k'^{O(1)}n^{O(1)}$, which is excluded by the conjecture.

Related Work. Guillemot [17] introduced the class WNL (which equals XNLP closed under fpt-reductions), and showed some problems to be complete for WNL, including a version of Longest Common Subsequence. The class XNLP (under a different name) was introduced by Elberfeld et al. [12], who also showed a number of problems, including Linear Cellular Automaton Acceptance, to be complete for the class. A large number of parameterized problems was shown to be XNLP-complete recently by Bodlaender et al. [4]. Very recently, in work that aims at separating the complexity of treewidth and pathwidth at one side, and stable gonality at another side, Bodlaender et al. [3] showed a number of flow problems parameterized by pathwidth to be complete for XNLP.

2 Overview of the results

In this section, we give a bird's-eye view of the results proved in this paper, and discuss related work for the specific problems we consider. Due to space limitations, statements marked with ♣ had their proofs deferred to the full version of this work.

Parameterized by linear clique-width. We consider the Max Cut, the Coloring, and the Maximum Regular Induced Subgraph problems parameterized by linear clique-width. Let $E(V_1, V_2)$ denote the set of edges with one endpoint in V_1 and one endpoint in V_2.

Max Cut
Input: A graph $G = (V, E)$ described by a given linear k-expression describing G and an integer W.
Parameter: k.
Question: Is there a bipartition of V into (V_1, V_2) such that $|E(V_1, V_2)| \geq W$?

In 1994, Wanke [25] showed that Max Cut is in XP for graphs of bounded NLC-width, which directly implies XP-membership with clique-width as parameter, as NLC-width and clique-width are linearly related. In 2014, Fomin et al. [14] consider the fine grained complexity for Max Cut for graphs of small clique-width, giving an algorithm with improved running time and showing asymptotic optimality (assuming the Exponential Time Hypothesis). From their results, it follows that Max Cut is $W[1]$-hard with clique-width as parameter. In Section 4.1, we prove the following theorem.

▶ **Theorem 3.** Max Cut *with linear clique-width as parameter is XNLP-complete.*

Next, we consider the classical Coloring problem, which given a graph G and an integer k asks if G has a proper coloring with k colors. Similarly to the story of the Max Cut problem, Coloring parameterized by clique-width was shown to be in XP by Wanke in 1994 [25], and a W[1]-hardness proof only followed in 2010 by Fomin et al. [13]. The XP algorithm for coloring runs in time $n^{O(2^k)}$, where k is the clique-width, and Fomin et al. [15] even showed that this run time can probably not be substantially improved: an algorithm running in time $n^{2^{o(k)}}$ would refute the ETH. We prove the following.

▶ **Theorem 4 (♣).** *Coloring parameterized by linear clique-width is XNLP-complete.*

Lastly, we consider the Maximum Regular Induced Subgraph problem. The problem was studied by several authors, including Asahiro et al. [1], who show among others an algorithm that uses linear time for graphs of bounded treewidth, where the time depends single exponentially on the treewidth. Moser and Thilikos [22], and independently Mathieson and Szeider [21] show (amongst other results) that the problem is $W[1]$-hard when the size of the subgraph (parameter W in our description below) is used as parameter. Broersma et al. [7] give XP algorithms for several problems, including MAXIMUM REGULAR INDUCED SUBGRAPH for graphs of bounded clique-width.

MAXIMUM REGULAR INDUCED SUBGRAPH
Input: A graph G described by a given linear k-expression and two integers W and D.
Parameter: k.
Question: Is there a D-regular induced subgraph of G on at least W vertices?

We show the following.

▶ **Theorem 5 (♣).** *MAXIMUM REGULAR INDUCED SUBGRAPH parameterized by linear clique-width is XNLP-complete.*

Parameterized by pathwidth. We consider the CAPACITATED RED-BLUE DOMINATING SET and CAPACITATED DOMINATING SET problems. Below, we give the formal statement of the problems, where we have the width of the path decomposition as parameter. One of the reasons of interest in these problems is that they model facility location problems: the red vertices model possible facilities that can serve a bounded number of clients which are modelled by the blue vertices.

CAPACITATED RED-BLUE DOMINATING SET
Input: A bipartite graph $G = (R, B, E)$, a path decomposition of G of width ℓ, a capacity function $c : R \to \mathbb{N}$, and an integer k.
Parameter: ℓ.
Question: Is there a subset S of R, and an assignment of blue vertices $f : B \to S$ such that $\{w, f(w)\} \in E$ for all $w \in R$ and $|f^{-1}(v)| \leq c(v)$ for all v in S?

CAPACITATED DOMINATING SET
Input: A graph $G = (V, E)$, a path decomposition of G of width ℓ, a capacity function $c : V \to \mathbb{N}$, and an integer k.
Parameter: ℓ.
Question: Is there a subset S of V, and an assignment of the vertices $f : V \to S$ such that $\{w, f(w)\} \in E$ or $w = f(w)$ for all $w \in V$ and $|f^{-1}(v)| \leq c(v)$ for all v in S?

In 2008, Dom et al. [10] showed that CAPACITATED DOMINATING SET is $W[1]$-hard, with the treewidth and solution size k as combined parameter. CAPACITATED DOMINATING SET was shown to be $W[1]$-hard for planar graphs, with the solution size as parameter by Bodlaender et al. [5]. Fomin et al. [14] give bounds for the fine grained complexity of CAPACITATED RED-BLUE DOMINATING SET, for graphs with a small feedback vertex set; their results imply that the problem is $W[1]$-hard with feedback vertex set as parameter. The proof of the following theorem can be found in Section 4.2.

▶ **Theorem 6.** CAPACITATED RED-BLUE DOMINATING SET *and* CAPACITATED DOMINATING SET *parameterized by pathwidth are XNLP-complete.*

Parameterized by logarithmic linear clique-width. Bodlaender et al. [4] introduced the parameter *logarithmic pathwidth* as $\mathbf{pw} / \log n$ for an n-vertex graph of pathwidth $\mathbf{pw}$. This allows the pathwidth to be linear in the logarithm of the number of vertices of the graph. Here we introduce the *logarithmic linear clique-width* as $\mathbf{lcw} / \log n$ for graphs on n vertices with linear clique-width $\mathbf{lcw}$.

We provide new XNLP-complete problems for the parameter logarithmic pathwidth, and show that these problems and the previously known XNLP-complete problems for this parameter [4] are also complete for the parameter logarithmic linear clique-width. Our results are summarised below.

The motivation to study the logarithmic linear clique-width or logarithmic pathwidth comes from the observation that many FPT algorithms with linear cliquewidth or pathwidth as parameter have a single exponential time dependency on the parameter. Thus, if linear cliquewidth or pathwidth is logarithmic in the size of the graph, these algorithms turn into XP algorithms.

▶ **Theorem 7** (♣). *When parameterized by logarithmic pathwidth or logarithmic linear clique-width,* Independent Set*,* Dominating Set*, q-*List-Coloring *for $q > 2$, and* Odd Cycle Transversal *are XNLP-complete, and* Feedback Vertex Set *is XNLP-hard.*

Lokshtanov et al. [20] established (tight) lower bounds for these problems for the parameter pathwidth under the Strong Exponential Time Hypothesis. Several of our gadgets are based on those used for these lower bounds by [20].

Parameterized by linear mim-width. We prove that several fundamental graph problems are XNLP-complete when parameterized by the mim-width of a given linear order of the input graph. W[1]-hardness for Independent Set and Dominating Set in this parameterization was shown by Fomin et al. [16], and for Feedback Vertex Set by Jaffke et al. [19]. For q-Coloring, $W[1]$-hardness was not known before our work. We would like to point out that our XNLP-hardness proof uses a gadget that requires five colors to construct, and it would be interesting to see if this can be improved to three colors. In section Section 4.3 we prove the result below for q-Coloring. The proofs for the remaining problems are deferred to the full version.

▶ **Theorem 8** (♣). *When parameterized by linear mim-width,* Independent Set*,* Dominating Set*, q-*Coloring *for any fixed $q \geq 5$ and* Feedback Vertex Set *are XNLP-complete.*

Bipartite bandwidth. We consider the following bipartite variant of the Bandwidth problem.

Bipartite Bandwidth
Input: A bipartite graph $G = (X, Y, E)$ and an integer k.
Parameter: k.
Question: Are there orderings $\alpha : X \to [n]$ and $\beta : Y \to [m]$ such that for each $uv \in E$, $|\alpha(u) - \beta(v)| \leq k$?

A possible application of this problem is as follows. Let a matrix M be given. Create a vertex $x_i \in X$ for each row i and a vertex $y_j \in Y$ for each column j, and let x_i be adjacent to y_j if and only if $M_{i,j} \neq 0$. This graph has bipartite bandwidth at most k if and only if the rows and columns of M can be permuted (individually) in such a way that all non-zero entries are within k distance from the main diagonal. We show the following.

▶ **Theorem 9** (♣). Bipartite Bandwidth *is XNLP-complete for trees.*

3 Preliminaries

The required background on the computational problems studied in this paper are given in their respective sections. The notions relevant to the entire paper are defined below.

We write $[n] = \{1, \ldots, n\}$ and $[a, b]$ for the set of integers x with $a \leq x \leq b$. All logarithms in this paper have base 2. We use $\mathbb{N}$ for the set of the natural numbers $\{0, 1, 2, \ldots\}$, and $\mathbb{Z}^+$ denotes the set of the positive natural numbers $\{1, 2, \ldots\}$. We write $N(S)$ and $N[S] = N(S) \cup S$ for the open and closed neighborhood of S.

3.1 Definition of the class XNLP

In this paper, we study parameterized decision problems, which are subsets of $\Sigma^* \times \mathbb{N}$, for a finite alphabet Σ. We assume the reader to be familiar with notions from parameterized complexity, such as XP, $W[1]$, $W[2]$, …, $W[P]$ (see e.g. [11]). The class XNLP (denoted $N[f\text{ poly}, f\log]$ by [12]) consists of the parameterized decision problems that can be solved by a non-deterministic algorithm that simultaneously uses at most $f(k)n^c$ time and at most $f(k)\log n$ space, on an input (x, k), where x can be denoted with n bits, f a computable function, and c a constant. We assume that functions f of the parameter in time and resource bounds are computable – this is called *strongly uniform* by Downey and Fellows [11]. More information about the complexity class XNLP can be found in [4].

3.2 Reductions

In the remainder of the paper, unless stated otherwise, completeness for XNLP is with respect to pl-reductions, which are defined below. The definitions are based upon the formulations in [12].

- A *parameterized reduction* from a parameterized problem $Q_1 \subseteq \Sigma_1^* \times \mathbb{N}$ to a parameterized problem $Q_2 \subseteq \Sigma_2^* \times \mathbb{N}$ is a function $f : \Sigma_1^* \times \mathbb{N} \to \Sigma_2^* \times \mathbb{N}$, such that the following holds.
  1. For all $(x, k) \in \Sigma_1^* \times \mathbb{N}$, $(x, k) \in Q_1$ if and only if $f((x, k)) \in Q_2$.
  2. There is a computable function g, such that for all $(x, k) \in \Sigma_1^* \times \mathbb{N}$, if $f((x, k)) = (y, k')$, then $k' \leq g(k)$.
- A *parameterized logspace reduction* or *pl-reduction* is a parameterized reduction for which there is an algorithm that computes $f((x, k))$ in space $O(g(k)+\log n)$, with g a computable function and $n = |x|$ the number of bits to denote x.

3.3 Pathwidth, linear clique-width, and linear mim-width

A *path decomposition* of a graph $G = (V, E)$ is a sequence $(X_1, X_2, \ldots, X_r)$ of subsets of V with the following properties.

1. $\bigcup_{1 \leq i \leq r} X_i = V$.
2. For all $\{v, w\} \in E$, there is an $i \in I$ with $v, w \in X_i$.
3. For all $1 \leq i_0 < i_1 < i_2 \leq r$, $X_{i_0} \cap X_{i_2} \subseteq X_{i_1}$.

The *width* of a path decomposition $(X_1, X_2, \ldots, X_r)$ equals $\max_{1 \leq i \leq r} |X_i| - 1$, and the *pathwidth* **pw** of a graph G is the minimum width of a path decomposition of G.

A k*-labeled* graph is a graph $G = (V, E)$ together with a labeling function $\Gamma : V \to [k]$. A k*-expression* constructs a k-labeled graph by the means of the following operations:

1. *Vertex creation*: $i(v)$ is the k-labeled graph consisting of a single vertex v which is assigned label i.
2. *Disjoint union*: $H \oplus G$ is the disjoint union of k-labeled graphs H and G.

3. *Join*: $\eta_{i\times j}(G)$ is the k-labeled graph obtained by adding all possible edges between vertices with label i and vertices with label j to G.
4. *Renaming label*: $\rho_{i\to j}(G)$ is the k-labeled graph obtained by assigning label j to all vertices labelled i in G.

A *linear* k-expression is a k-expression with the additional condition that one of the arguments of the disjoint union operation needs to be a graph consisting of a single vertex. The *clique-width* $\mathbf{cw}(G)$(resp. *linear clique-width* $\mathbf{lcw}(G)$) of a graph G is the minimal k such that G can be constructed by a k-expression (resp. linear k-expression) with any labeling.

For a graph $G = (V, E)$ and $A, B \subseteq V$ with $A \cap B = \emptyset$, we let $G[A, B]$ be the bipartite subgraph of G with vertices $A \cup B$ and edges $\{ab \mid ab \in E, a \in A, b \in B\}$. We let $\text{cutmim}_G(A, B)$ be the size of a maximum induced matching in $G[A, B]$ and $\text{mim}_G(A) = \text{cutmim}_G(A, V \setminus A)$. Here, an induced matching $M \subseteq E$ is a matching such that there are no additional edges between the endpoints of M in the graph in question. The *mim-width* of a linear order $v_1, \ldots, v_n$ of V is the maximum, over all i, of $\text{mim}_G(\{v_1, \ldots, v_i\})$. The *linear mim-width* of G is the minimum mim-width over all linear orders of V.

3.4 Chained variants of Satisfiability and Multicolored Clique

In [4], the following problems were introduced, and shown to be XNLP-complete.

CHAINED POSITIVE CNF-SAT
Input: r sets of Boolean variables $X_1, X_2, \ldots X_r$, each of size q; an integer $k \in \mathbb{N}$; Boolean formula ϕ, which is in conjunctive normal form and an expression on $2q$ variables, using only positive literals; for each i, a partition of X_i into $X_{i,1}, \ldots, X_{i,k}$ such that $\forall j, j' \in [k], |X_{i,j}| = |X_{i,j'}|$.
Parameter: k.
Question: Is it possible to satisfy the formula

$$\bigwedge_{1\leq i\leq r-1} \phi(X_i, X_{i+1})$$

by setting from each set $X_{i,j}$ exactly 1 variable to true and all others to false?

CHAINED MULTICOLORED CLIQUE
Input: Graph $G = (V, E)$, partition of V into $V_1, \ldots, V_r$, such that for each edge $uv \in E$ with $u \in V_i$ and $v \in V_j$, $|i - j| \leq 1$, function $f : V \to [k]$.
Parameter: k.
Question: Is there a set $W \subseteq V$ such that for all $i \in [r-1]$, $W \cap (V_i \cup V_{i+1})$ is a clique, and for each $i \in [r]$ and $j \in [k]$, there is a vertex $v \in W \cap V_i$ with $f(v) = j$?

The CHAINED MULTICOLORED INDEPENDENT SET problem is defined analogously, with the only difference that the solution W is required to be an independent set.

▶ **Theorem 10** (Bodlaender et al. [4]). *CHAINED POSITIVE CNF-SAT, CHAINED MULTICOLORED CLIQUE and CHAINED MULTICOLORED INDEPENDENT SET are XNLP-complete.*

4 Problems parameterized by linear width measures

In this section we prove XNLP-completeness for three of the problems mentioned in Section 2 parameterized by linear width measures. The full version of this work contains all the proofs of the results stated in Section 2.

4.1 Max Cut parameterized by linear clique-width

In this section, we consider the MAX CUT problem, with the linear clique-width as parameter, and show it to be XNLP-complete. Our result is based upon the XNLP-hardness result for a problem, called CIRCULATING ORIENTATION, with pathwidth as parameter. Borrowing from terminology from flows in graphs, we say that a directed graph $G = (V, A)$ with for each arc $a \in A$ a weight $w(a) \in \mathbf{N}$, is a *circulation*, if for each vertex v, the total weight of all incoming arcs at v equals the total weight of all arcs outgoing from v. We reduce from the following problem.

CIRCULATING ORIENTATION
Input: An undirected graph $G = (V, E)$ with a path decomposition of G of width ℓ, an edge weight function $w : E \to \mathbb{N}$, given in unary notation.
Parameter: ℓ.
Question: Is there an orientation of G that is a circulation?

▶ **Theorem 11** (Bodlaender et al. [3]). CIRCULATING ORIENTATION *is XNLP-complete.*

▶ **Theorem 3.** MAX CUT *with linear clique-width as parameter is XNLP-complete.*

Proof. We first show membership in XNLP. The main idea is to turn the existing dynamic programming that solves the problem given a k-expression of an n-vertex graph of linear clique-width k into a non-deterministic algorithm, by guessing an element from a table instead of building full tables. For each vertex creation, we guess on which side of the partition the vertex is. We maintain the following certificate: for each label, the number of vertices on each side of the bipartition, and the number of edges of the current expression that were in the cut. Since there are at most k labels and the size of the cut is bounded by the number of edges, this certificate uses only $O(k \log n)$ bits.

To show hardness for XNLP, we reduce from CIRCULATING ORIENTATION with pathwidth as parameter. Suppose we have an instance for CIRCULATING ORIENTATION: an undirected graph G with edge weight function w. For each vertex v, write $D(v)$ as the total weight of all edges incident to v.

We build a new, undirected graph $H = (V_H, E_H)$ as follows. For each vertex $v \in V$, each edge e with v as one of its endpoints, and each integer $i \in [1, w(e)]$, we create a vertex $x_{v,e,i}$. Two distinct vertices $x_{v,e,i}$ and $x_{w,e',j}$ are adjacent if and only if $v = w$ or $e = e'$. In other words: for each vertex $v \in V$, we have a clique with $D(v)$ vertices, which consists of all vertices of the form $x_{v,\cdot,\cdot}$, that we call *the clique of* v. For each edge $e = \{v, w\} \in E$, we have a clique with $2w(e)$ vertices, namely all vertices of the form $x_{v,e,\cdot}$ and $x_{w,e,\cdot}$. See Figure 1 for a partial example.

▷ Claim 12. G has a circulating orientation if and only if H has a bipartition that cuts $\sum_{e \in E} w(e)^2 + \sum_{v \in V} D(v)^2/4$ edges.

Proof. Suppose G has a circulating orientation. For each edge $e = \{v, w\}$, if the orientation directs v to w, then add all vertices of the form $x_{v,e,i}$ to Z_1 and all vertices of the form $x_{w,e,i}$ to Z_2 ($i \in [1, w(e)]$); otherwise, add all vertices of the form $x_{v,e,i}$ to Z_2 and all vertices of the form $x_{w,e,i}$ to Z_1 ($i \in [1, w(e)]$).

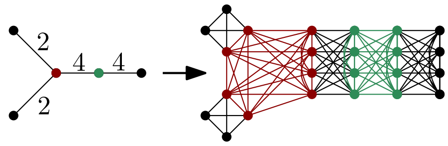

Figure 1 Example for the construction of the hardness proof of Max Cut (fragment).

Since we started from a circulating orientation, for each vertex v there are $D(v)/2 \times D(v)/2$ edges of the form $\{x_{v,\cdot,\cdot}, x_{v,\cdot,\cdot}\}$ crossing the bipartition. Moreover, there are $w(e) \times w(e)$ edges of the form $\{x_{v,e,\cdot}, x_{w,e,\cdot}\}$ crossing the bipartition for each edge $e = \{v, w\}$. We conclude that the bipartition cuts the required number of edges.

Now, suppose we have a partition Z_1, Z_2 of V_H with $\sum_{e \in E} w(e)^2 + \sum_{v \in V} D(v)^2/4$ edges between Z_1 and Z_2. We distinguish two types of edges in $E_H \cap (Z_1 \times Z_2)$. A Type 1 edge is an edge between two vertices $x_{v,e,i}$ and $x_{v,e',j}$ (i.e., it is in the clique of a vertex v). A Type 2 edge is an edge between two vertices $x_{v,e,i}$ and $e_{w,e,j}$ for some edge $e = \{v, w\}$. Note that each edge in H is of Type 1 or Type 2 and that H has precisely $\sum_{e \in E} w(e)^2$ Type 2 edges.

For each vertex $v \in V$, we consider how many Type 1 edges (those in the clique of v) are in $Z_1 \times Z_2$. If we have α vertices in the clique of v that belong to Z_1, then $D(v) - \alpha$ vertices in the clique of v belong to Z_2, and thus, in this clique, we cut $\alpha \cdot (D(v) - \alpha) \leq D(v)^2/4$ edges; the maximum possible is reached when $\alpha = D(v)/2$.

It follows that the number of Type 1 edges that are cut is at most $\sum_{v \in V} D(v)^2/4$. So, we must cut all Type 2 edges, i.e., for each edge $e = \{v, w\}$, all edges of the form $\{x_{v,e,i}, x_{w,e,j}\}$ are between a vertex in Z_1 and a vertex in Z_2. It follows that we either have that all vertices of the form $x_{v,e,i}$ are in Z_1 and all vertices of the form $x_{w,e,i}$ are in Z_2 – in which case we direct the edge e from v to w; or all vertices of the form $x_{v,e,i}$ are in Z_2 and all vertices of the form $x_{w,e,i}$ are in Z_1, and now we direct the edge from w to v.

For each vertex $v \in V$, we must have exactly $D(v)/2$ vertices from the clique of v in Z_1 and equally many vertices in Z_2; otherwise, we cannot reach the required number of cut edges. Now, the total weight of all edges that we directed out of v precisely equals the number of vertices in the clique of v in Z_1, and similarly, the total weight of all edges that we directed towards of v precisely equals the number of vertices in the clique of v in Z_2. Both numbers equal $D(v)/2$. As this holds for each $v \in V$, the orientation defined above is a circulation. ◁

Finally, we show that we can construct a linear clique expression for H given a path decomposition of G; the number of colors we use for the clique width construction equals the width of the path decomposition plus 4. The construction uses ideas for constructing clique width constructions for line graphs of graphs of bounded treewidth; see [18].

Suppose we have a nice path decomposition, which uses introduce vertex, introduce edge, and forget nodes. We use $k + 1$ active colors – each active color will correspond to one vertex in the current bag. We also have an inactive color, which we will denote by the letter o. We also use two temporary colors, which we call α and β.

We sequentially visit the bags of the path decomposition. Bags correspond to a number of steps of the construction of H, as described next. If we introduce a vertex, we select a currently unused active color, and say this is the color of that vertex, and assume it to be used. If we introduce an edge $e = \{v, w\}$, we add the vertices $x_{v,e,i}$ one by one, each with the color α. Then, we add the vertices $x_{w,e,i}$ one by one, each with the color β. Now, we add all edges between vertices of color α and β. Now, recolor all vertices of color α by the color

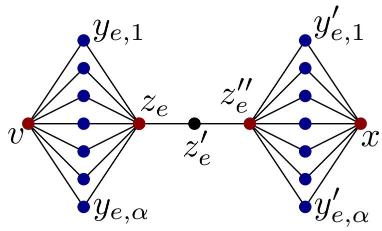

Figure 2 Edge gadget from the proof of Theorem 6.

of v. Then, recolor the vertices of color β by the color of w. If we forget a vertex v, we first add edges between all vertices of the color of v – at this point, these are all vertices in the clique of v, thus effectively ensuring that this set of vertices indeed is a clique. Then, recolor the vertices with the color of v with the inactive color o. Consider the color of v now unused.

One can verify that this indeed constructs precisely H, and that the construction can be done with $f(k)\log n$ additional space. ◀

4.2 Variants of Dominating Set parameterized by pathwidth

In this section we prove the following theorem.

▶ **Theorem 6.** Capacitated Red-Blue Dominating Set *and* Capacitated Dominating Set *parameterized by pathwidth are XNLP-complete.*

Proof. We first show membership in XNLP for Capacitated Red-Blue Dominating Set. For each red vertex, we guess if it is in the dominating set, and for each edge from a chosen red vertex to a blue neighbor, we guess if it is used for dominating. We do this while going through the path decomposition from left to right. We need to keep track which blue vertices are already dominated, which red vertices are in the dominating set plus their remaining capacity, and the total number of vertices in the dominating set so far. We may assume that the remaining capacities are never larger than the number of blue vertices; therefore, we only need to store $O(\log n)$ bits per vertex in the current bag. Membership in XNLP follows in a similarly for Capacitated Dominating Set.

Hardness follows by a reduction from Circulating Orientation (defined in Section 4.1). Suppose that we are given an input of Circulating Orientation, say a graph $G = (V, E)$ with weight function $w : E \to \mathbb{N}$. We assume that these weights are given in unary. Note that in a solution, the total weight of edges directed towards a vertex v and the total weight of the edges directed out of v should both equal $\sum_{\{v,x\}\in E} w(\{v, x\})/2$.

We build a graph as follows. For each vertex $v \in V$, we create a vertex v, colored red, in H. We give v a private blue neighbor v'. The capacity of v equals $1 + \sum_{\{v,x\}\in E} w(\{v, x\})/2$. We can assume this capacity is integral, otherwise there is no solution to the instance (G, w). Each edge $e = \{v, x\} \in E$ is replaced by the following gadget. Suppose $w(\{v, x\}) = \alpha \in \mathbb{N}$. We create $2\alpha + 3$ vertices, called $y_{e,1}, y_{e,2}, \ldots, y_{e,\alpha}$, z_e, z'_e, z''_e, $y'_{e,1}, \ldots, y'_{e,\alpha}$. The edge e is replaced by the subgraph shown in Figure 2. The vertices z_e and z''_e are red, and all other new vertices are blue. We give the new red vertices z_e and z'''_e a capacity that equals their degree.

Let $H = (V_H, E_H)$ be the resulting red-blue colored graph, with $c(v)$ the capacity of a red vertex $v \in V_H$. We claim that H has a dominating set of size size $|V| + |E|$ for which each chosen red vertex dominates at most its capacity many blue vertices, if and only if G has a circulating orientation.

Suppose first that we have a set S of red vertices with $|S| \leq |V| + |E|$, and an assignment of blue vertices to neighbors in S, such that no red vertex has more than its capacity number of vertices assigned to it.

Each vertex that is a copy of a vertex from V must belong to S, as they have a private blue neighbor. For each edge e, either z_e or z''_e must be in S, to dominate z'_e. This gives in total already $|V| + |E|$ vertices, so no edge can have both z_e and z''_e in S. For each edge $e = \{v, x\}$, if $z_e \in S$, then orient the edge from v to x in G; if $z'''_e \in S$, then orient the edge from x to v. Now, for each $v \in V$, the total weight of incoming edges of the orientation can be at most $c(v) - 1$, since v must also dominate its private neighbor. By definition, $c(v) - 1 = \sum_{\{v,x\} \in E} w(\{v, x\})/2$. This means that for each vertex, the total weight of incoming edges is at most half the total weight of incident edges; it follows that this total weight must be equal, because when there is a vertex for which this weight is smaller, then there must be another vertex for which it is larger. So, we have an orientation that is a circulation.

Suppose now that we have a circulation that is an orientation. Add each original vertex $v \in V$ to S, and for each edge $e = \{v, x\}$, place z_e in S when the edge is oriented from v to x and otherwise place z''_e in S. Red vertices on edge gadgets dominate all their neighbors; red original vertices dominate their private neighbor and all not yet dominated blue neighbors. This gives a dominating set where each red vertex in S dominates precisely its capacity many neighbors, as desired.

Finally, we show that we can build a log-space transducer that transforms a path decomposition of G of width ℓ to one of H with width at most $\ell + 2$. We first ensure that the path decomposition of G is nice (which can be done via a log-space transducer). We pass through the bags from left to right. For a forget bag in the path decomposition of G, we take the same bag for H. For an introduce bag $X_i = X_{i-1} \cup \{v\}$, we loop through the vertices in X_{i-1} one-by-one, say these are $x_1, \ldots, x_r$. For each $j \in [r]$, if $\{v, x_j\} \in E$, then we add the following bags (in order):

$$X_i \cup \{z_e, y_{e,1}\},\ X_i \cup \{z_e, y_{e,2}\}, \ldots, X_i \cup \{z_e, y_{e,w(e)}\},\ X_i \cup \{z_e, z'_e\},\ X_i \cup \{z'_e, z''_e\},$$
$$X_i \cup \{z''_e, y'_{e,1}\},\ X_i \cup \{z''_e, y'_{e,2}\}, \ldots,\ X_i \cup \{z''_e, y'_{e,w(e)}\}.$$

One can verify that this gives a path decomposition of H. The width has increased by at most 2.

A standard transformation now shows that Capacitated Dominating Set is also XNLP-hard with pathwidth as parameter. Given an instance (G, w) of Capacitated Red-Blue Dominating Set, we build an equivalent instance of Capacitated Dominating Set. We give each blue vertex capacity zero. We add two new vertices x and x', with x' of degree one and x adjacent to all red vertices and to x'. The capacity of x is equal to the number of red vertices plus 2. We increase the target size of the solution by one (and remove all colors). The pathwidth has gone up by at most one. ◀

We remark that a similar reduction can be used to show XNLP-hardness of Capacitated Vertex Cover (by removing the vertex z'_e, having parallel paths of length 3 instead of 2 in the gadget of Figure 2 and giving each original vertex a new neighbor of degree one).

4.3 q-Coloring parameterized by linear mim-width

In this section we show that for each fixed $q \geq 5$, q-Coloring is XNLP-complete when parameterized by the linear mim-width of the input graph.

> q-Coloring
> **Input:** A graph $G = (V, E)$ and a linear order of V of mim-width w.
> **Parameter:** w.
> **Question:** Does G have a proper vertex-coloring with q colors?

XNLP-membership for this problem will be shown via the corresponding known dynamic programming XP-algorithm [8] which is based on the following.

▶ **Definition 13** (Neighborhood Equivalence). *Let $G = (V, E)$ be a graph and $A \subseteq V$. For all $X, Y \subseteq A$: $X \equiv_A Y \Leftrightarrow N(X) \cap (V \setminus A) = N(Y) \cap (V \setminus A)$.*

▶ **Lemma 14.** *q-Coloring parameterized by the mim-width of a linear order of the vertices of the input graph is in XNLP.*

Proof. Let n be the number of vertices of the input graph $G = (V, E)$ and w the mim-width of the given linear order $v_1, \ldots, v_n$ of V. Membership in XNLP is shown by adapting the XP-algorithm [8] to a nondeterministic polynomial-time algorithm that also uses at most $O(w \log n)$ space.

With i going from 1 to n, at step i we store partial solutions associated with the subgraph of G induced by the vertices $V_i = \{v_1, \ldots, v_i\}$. (For convenience, we let $\overline{V_i} = V \setminus V_i$.) In the XP algorithm of [8], partial solutions are proper colorings of $G[V_i]$ and a table index consists of representatives of equivalence classes $\mathcal{Q}_1, \ldots, \mathcal{Q}_q$ of $\equiv_{V_i}$ such that for all $i \in [q]$, color class i in the coloring is contained in $\mathcal{Q}_i$.

To prove that each such coloring can be represented using $O(w \log n)$ bits, we use the following claim shown in [8]. We reprove it here to clarify that it leads to an algorithm satisfying the time and space requirements.

▷ Claim 15. For each $i \in [n-1]$, and each $S_i \subseteq V_i$, there is a set $R_i \subseteq V_i$ with $R_i \equiv_{V_i} S_i$ and $|R_i| \leq w$. Furthermore, there is a polynomial-time algorithm using at most $O(w \log n)$ space that determines R_i from R_{i-1}, where $R_{i-1} \equiv_{V_{i-1}} S_i \cap V_{i-1}$ and $|R_{i-1}| \leq w$.

Proof. For $i \leq w$, we can simply let $R_i = S_i$, so suppose that $i > w \geq 1$, and that $|S_i| > w$. By induction, we can assume that we have $R_{i-1} \subseteq V_{i-1}$ of size at most w such that $R_{i-1} \equiv_{V_{i-1}} S_i \cap V_{i-1}$. Let $R'_i = R_{i-1} \cup \{v\}$. If $|R'_i| \leq w$, then we let $R_i = R'_i$ and we are done. We may assume that $|R'_i| = w + 1$. If there is some $x \in R'_i$ such that $N(R'_i \setminus \{x\}) \cap \overline{V_i} = N(R'_i) \cap \overline{V_i}$, then we let $R_i = R'_i \setminus \{x\}$ and we are done. Otherwise, we know that each vertex x in R'_i has a neighbor y in $\overline{V_i}$ such that y is non-adjacent to all vertices in $R'_i \setminus \{x\}$. This means that these xy-edges form an induced matching in $G[V_i, \overline{V_i}]$, a contradiction. ◁

The previous claim immediately shows that each table index can be encoded using $O(qw \log n)$ bits, which is $O(w \log n)$ since q is a constant. The algorithm works as follows. Upon arrival of the next vertex v_{i+1}, we nondeterministically guess which of the q colors v_{i+1} receives. We then nondeterministically guess the table index corresponding to the updated solution. By Claim 15 we can conclude that the nondeterministic step can be implemented in polynomial time, and using only $O(w \log n)$ space. ◀

For two disjoint sets $A, B \subseteq V$, the *bipartite complement* replaces each edge with one endpoint in A and the other in B with a non-edge and vice versa. The following construction is due to Fomin et al. [16].

▶ **Definition 16** (Subdivision-complement, Fomin et al. [16]). *Let $G = (V, E)$ be a graph, and $A, B \subseteq V$ with $A \cap B = \emptyset$. The* subdivision-complement between A and B *is the following operation:*

1. *Subdivide each edge uv with $u \in A$ and $v \in B$; call the resulting set of vertices R.*
2. *Take the bipartite complement between A and R and the bipartite complement between B and R.*

The reason why this operation is useful for reductions for problems parameterized by mim-width are the following bounds on the maximum induced matching size of cuts resulting from this construction. This can also be derived from [16], but we include a simple direct proof here for completeness.

▶ **Lemma 17.** *Let $G = (V, E)$ be a graph, and $A, B \subseteq V$ with $A \cap B = \emptyset$. Let G' be the graph obtained from G by applying the subdivision-complement between A and B; let R denote the set of vertices created in the construction. Then, for all $C \in \{A, B\}$, $\mathrm{cutmim}_{G'}(C, R) \leq 2$.*

Proof. Suppose for a contradiction that there is an induced matching of size three in $G'[A, R]$, say $M = \{a_i r_i \mid i \in [3], a_i \in A, r_i \in R\}$. For all $i \in [3]$, let e_i denote the edge in G whose subdivision created vertex r_i. Since M is an induced matching and by construction, a_1 is the endpoint of e_2 and e_3. But this implies that a_2 is not the endpoint of e_3, and therefore that the edge $a_2 r_3$ exists in $G'[A, R]$. ◀

To prove the bound on the mim-width of linear orders constructed in the hardness proofs in this section, we need the following additional lemma which can be seen as a variation of a lemma in [6], but for linear mim-width. Recall that for a graph $G = (V, E)$ and a partition $\mathcal{P}$ of V, the *quotient graph $G/\mathcal{P}$* is the graph obtained from G by contracting each part of $\mathcal{P}$ into a single vertex. The *cutwidth* of a linear order $\Lambda = v_1, \ldots, v_n$ of V, denoted by $\mathrm{cutw}(\Lambda)$ is the maximum, over all i, of the number of edges with one endpoint in $\{v_1, \ldots, v_i\}$ and the other in $\{v_{i+1}, \ldots, v_n\}$.

▶ **Lemma 18** (♣). *Let $G = (V, E)$ be a graph, let $\mathcal{P} = (P_1, \ldots, P_r)$ be a partition of V, and let $G' = G/\mathcal{P}$. For all $i \in [r]$ let Λ_i be a linear order of P_i such that $\mathrm{mimw}_{G[P_i]}(\Lambda_i) \leq c$, and suppose that for all distinct $i, j \in [r]$, $\mathrm{cutmim}_G(P_i, P_j) \leq d$. Let $\Lambda = \Lambda_1, \Lambda_2, \ldots, \Lambda_r$, and let $\Lambda' = P_1, \ldots, P_r$ be the corresponding linear order of $G/\mathcal{P}$. Then, $\mathrm{mimw}(\Lambda) \leq 2d \cdot \mathrm{cutw}(\Lambda') + c$.*

▶ **Definition 19** (Frame graph). *Let $(G = (V, E), V_1, \ldots, V_r, f)$ be an instance of* Chained Multicolored Clique*; for each $i \in [r]$, let $V(i, 1), \ldots, V(i, k)$ denote the partition of V_i according to f.*

The frame graph $G' = (V', E')$ *is obtained from G by applying, for each $h \in [r-1]$ and each pair $(i_1, j_1), (i_2, j_2) \in \{h, h+1\} \times [k]$, where $(i_1, j_1) <_{LEX} (i_2, j_2)$, the subdivision-complement between $V(i_1, j_1)$ and $V(i_2, j_2)$.*[3] *We denote the set of new vertices by $R(i_1, j_1, i_2, j_2)$.*

For convenience, we let $\mathcal{P}$ denote the partition of V' into $V(1, 1)$, …, $V(r, k)$, $R(1, 1, 1, 1)$, …, $R(r, k, r, k)$, and we define the following auxiliary partial function $\phi \colon E \to V'$: For all (i_1, j_1) and (i_2, j_2) as above, for each $v_1 \in V(i_1, j_1)$ and $v_2 \in V(i_2, j_2)$ with $v_1 v_2 \in E$, we let $\phi(v_1 v_2) \in R(i_1, j_1, i_2, j_2)$ be the vertex created when subdividing $v_1 v_2$.

[3] Here, $<_{LEX}$ denotes the lexicographic ordering. We have $(i_1, j_1) <_{LEX} (i_2, j_2)$ if either $i_1 < i_2$ or if $i_1 = i_2$ and $j_1 < j_2$.

▶ **Lemma 20.** *Let $(G = (V, E), V_1, \dots, V_r, f)$ be an instance of* Chained Multicolored Clique*, and let $G' = (V', E')$ be its frame graph; adapt the notation from Definition 19. Then, G' has an independent set S with $|S \cap P| = 1$ for all $P \in \mathcal{P}$ if and only if G has a chained multicolored clique.*

Proof. Suppose G' has an independent set S with $|S \cap P| = 1$ for all $P \in \mathcal{P}$. Let $v_{i,j} \in S \cap V(i,j)$ for all $i \in [r]$, $j \in [k]$. We claim that this implies that for all $h \in [r-1]$, and all $(i_1, j_1), (i_2, j_2) \in \{h, h+1\} \times [k]$ with $(i_1, j_1) <_{LEX} (i_2, j_2)$, we have that $\phi(v_{i_1,j_1} v_{i_2,j_2}) \in S \cap R(i_1, j_1, i_2, j_2)$, which implies that $v_{i_1,j_1} v_{i_2,j_2} \in E$ and in particular that $S \cap V$ is a chained multicolored clique in G. Let $r \in S \cap R(i_1, j_1, i_2, j_2)$ and suppose $r \neq \phi(v_{i_1,j_1} v_{i_2,j_2})$. We may assume that $r = \phi(v, w)$ where $v \in V(i_1, j_1) \setminus \{v_{i_1,j_1}\}$. But then, $v_{i_1,j_1} r$ is an edge in G', a contradiction.

For the other direction, let $W \subseteq V$ be the chained multicolored clique in G. Let $S = \emptyset$. For each $i \in [r]$ and $j \in [k]$, we add the vertex $v_{i,j} \in W \cap V(i,j)$ to S. Next, for each $h \in [r-1]$, and each pair $(i_1, j_1), (i_2, j_2) \in \{h, h+1\} \times [k]$ where $(i_1, j_1) <_{LEX} (i_2, j_2)$, we add $\phi(v_{i_1,j_1} v_{i_2,j_2})$ to S. Note that since W is a chained multicolored clique, the edge $v_{i_1,j_1} v_{i_1,j_2}$ always exists in G. It follows immediately from the construction that S is an independent set in G', and that for all $P \in \mathcal{P}$, $|S \cap P| = 1$. ◀

▶ **Lemma 21.** *For fixed $q \geq 5$, q-*Coloring *parameterized by the mim-width of a given linear order of the vertices of the input graph is XNLP-hard.*

Proof. We give a parameterized logspace reduction from Chained Multicolored Clique to 5-List-Coloring. Let $\mathcal{I} = (G = (V, E), V_1, \dots, V_r, f)$ be the instance of Chained Multicolored Clique. We create the graph $G'' = (V'', E'')$ of the 5-List-Coloring instance as follows: Let $G' = (V', E')$ be the frame graph of $\mathcal{I}$ and adapt the notation of Definition 19. We obtain G'' and the lists $L'' = \{L(v) \mid v \in V''\}$ as follows:

- For each $P \in \mathcal{P}$, we add two vertices $a(P)$ and $b(P)$, and make $P'' = P \cup \{a(P), b(P)\}$ a path from $a(P)$ to $b(P)$. We let $\mathcal{P}'' = \{P'' \mid P \in \mathcal{P}\}$.
- For each $(i,j) \in [r] \times [k]$, each list of a vertex in $P = V(i,j)$ is $[3]$. If $|P|$ is even, the lists of both $a(P)$ and $b(P)$ are $\{1\}$; and if $|P|$ is odd, the list of $a(P)$ is $\{1\}$, and the list of $b(P)$ is $\{2\}$.
- For each $h \in [r-1]$ and $(i_1, j_1), (i_2, j_2) \in \{h, h+1\} \times [k]$ with $(i_1, j_1) <_{LEX} (i_2, j_2)$, each list of a vertex in $R = R(i_1, j_1, i_2, j_2)$ is $\{3, 4, 5\}$. If $|R|$ is even, the lists of both $a(R)$ and $b(R)$ are $\{5\}$; and if $|R|$ is odd, the list of $a(R)$ is $\{5\}$, and the list of $b(R)$ is $\{4\}$.

The following observation is immediate from the above construction.

▶ **Observation 22.** *In each proper list coloring of (G'', L'') and each $P \in \mathcal{P}$, there is a vertex in P that received color 3. Conversely, if some vertex $v \in P$ received color 3 in a proper list coloring of (G'', L''), then the vertices in $P'' \setminus \{v\}$ can be properly list-colored with colors $\{1, 2\}$, if $P = V(i,j)$ for some i, j or with colors $\{4, 5\}$, if $P = R(i_1, j_1, i_2, j_2)$, for some i_1, i_2, j_1, j_2.*

Now suppose that G has a chained multicolored clique W. Then by Lemma 20, there is an independent set S in G' such that $|S \cap P| = 1$ for all $P \in \mathcal{P}$. Note that S is also an independent set in G''. We can therefore let S be color class 3, and by Observation 22, the remaining vertices of each path P can be properly list colored without using color 3. It suffices to check the the edges between $V(i_1, j_1)$ and $R(i_1, j_1, i_2, j_2)$, for any valid choice of i_1, i_2, j_1, j_2. Since S is an independent set, the vertices $v \in S \cap V(i_1, j_1)$ and $w \in S \cap R(i_1, j_1, i_2, j_2)$ are non-adjacent. Furthermore, v has a different color from all vertices in $R(i_1, j_1, i_2, j_2) \setminus S$. Finally, the sets of colors appearing on $V(i_1, j_1) \setminus S$ and $R(i_1, j_1, i_2, j_2) \setminus S$ are disjoint.

Conversely, suppose that (G'', L'') has a proper list-coloring. Then we can combine Observation 22 and Lemma 20 to conclude that G has a chained multicolored clique (with color class 3 being the independent set required by Lemma 20).

We conclude with the following claim.

▷ Claim 23. One can in polynomial time and logarithmic space construct a linear order Λ of V'' such that $\text{mimw}(\Lambda) = O(k^2)$.

Proof. Each part $P \in \mathcal{P}$, together with $a(P)$ and $b(P)$ forms a path. Therefore we can trivially obtain a linear order of $P \cup \{a(P), b(P)\}$ by following the path from $a(P)$ to $b(P)$ whose mim-width is 1. For all $i \in [r]$ and $j \in [k]$, we let $\Lambda(i,j)$ be such a linear order where P corresponds to $V(i,j)$. For all $h \in [r-1]$, and all $(i_1, j_1), (i_2, j_2) \in \{h, h+1\} \times [k]$ with $(i_1, j_1) <_{LEX} (i_2, j_2)$, we let $\Gamma(i_1, j_1, i_2, j_2)$ be such a linear where P corresponds to $R(i_1, j_1, i_2, j_2)$. The desired linear order Λ traverses V'' as follows: Consider $(i,j) \in [r] \times [k]$ in lexicographically increasing order. First, we follow $\Lambda(i,j)$, and then $\Gamma(i,j,i,j+1)$, …, $\Gamma(i,j,i,k)$, and if $i < r$, then $\Gamma(i,j,i+1,1)$, …, $\Gamma(i,j,i+1,k)$. This linear order of G'' can be created using $O(\log n)$ bits of memory, where $n = |V|$.

As pointed out above, each $\Lambda(i,j)$ and each $\Gamma(i_1, j_1, i_2, j_2)$ has mim-width at most 1. The only edges in G' between different parts of $\mathcal{P}$ are between $V(i_1, j_1)$ and $R(i_1, j_1, i_2, j_2)$, and between $V(i_2, j_2)$ and $R(i_1, j_1, i_2, j_2)$, where $(i_1, j_1) <_{LEX} (i_2, j_2)$. By construction it therefore follows from Lemma 17 that for each pair of distinct parts $P_1, P_2 \in \mathcal{P}$, $\text{cutmim}_{G'}(P_1, P_2) \leq 2$. Let Λ' be the linear order of the vertices of $G''/\mathcal{P}$ such Λ can be obtained by traversing the parts of $\mathcal{P}$ in the order of Λ', and then following the above described order on each part of $\mathcal{P}$. We can observe that $\text{cutw}(\Lambda') = O(k^2)$, and therefore the claim follows from Lemma 18. ◁

We have shown that 5-List-Coloring parameterized by the mim-width of a given linear order of the input graph is XNLP-hard. To derive XNLP-hardness of 5-Coloring in the same parameterization, observe that we can use the standard trick of adding a clique on vertices $\{1, \ldots, 5\}$, and for each $i \in [5]$, connecting i and v if $i \notin L(v)$. Since adding c vertices can only increase the mim-width of a given linear order by at most c, no matter where the new vertices are placed, this does not prohibitively increase the linear mim-width either.

To obtain hardness for any $q > 5$, we simply add $q - 5$ universal vertices to the 5-Coloring instance obtained in the previous paragraph. Adding universal vertices cannot increase the mim-width w of any linear order, regardless of where they are placed, unless $w = 0$. ◀

Combining Lemmas 14 and 21, we obtain that q-Coloring is XNLP-complete parameterized by linear mim-width, for each fixed $q \geq 5$.

5 Conclusion

In this paper, we gave a number of XNLP-completeness proofs for graph problems parameterized by linear width measures. Such results are interesting for a number of reasons: they pinpoint the "right" complexity class for parameterized problems, they imply hardness for all classes $W[t]$, and they tell that it is unlikely that there is an algorithm that uses 'XP time and FPT space' by Conjecture 1.

This paper gives among others the first examples of XNLP-complete problems when the linear clique-width or linear mim-width is taken as parameter. Our hardness results give new starting points for future hardness and completeness proofs, in particular for problems with width measures like pathwidth, (linear) clique-width, or (linear) mim-width as parameter.

Other interesting directions for future research in this line are, for instance, to consider the parameterization by *cutwidth*; a promising candidate problem to show XNLP-completeness parametized by cutwidth is List Edge Coloring. Another interesting parameter to consider in this context is the *degeneracy* of a graph. It is also interesting to explore the concept of XNLP-completeness for width measures of other objects than graphs. One could for instance consider (linear) width measures of digraphs or hypergraphs.

We also leave open what the correct parameterized complexity class is for Feedback Vertex Set parameterized by logarithmic pathwidth or logaritmic linear cliquewidth – we showed XNLP-hardness, but did not prove containment in XNLP.

References

1. Yuichi Asahiro, Hiroshi Eto, Takehiro Ito, and Eiji Miyano. Complexity of finding maximum regular induced subgraphs with prescribed degree. *Theor. Comput. Sci.*, 550:21–35, 2014. doi:10.1016/j.tcs.2014.07.008.
2. Hans L. Bodlaender. Parameterized complexity of bandwidth of caterpillars and weighted path emulation. In Lukasz Kowalik, Michal Pilipczuk, and Pawel Rzazewski, editors, *Proceedings of the 47th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2021)*, volume 12911 of *Lecture Notes in Computer Science*, pages 15–27. Springer, 2021. doi:10.1007/978-3-030-86838-3_2.
3. Hans L. Bodlaender, Gunther Cornelissen, and Marieke van der Wegen. Problems hard for treewidth but easy for stable gonality. In Michael A. Bekos and Michael Kaufmann, editors, *Proceedings 48th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2022)*, volume 13453 of *Lecture Notes in Computer Science*, pages 84–97, 2022. doi:10.1007/978-3-031-15914-5_7.
4. Hans L. Bodlaender, Carla Groenland, Jesper Nederlof, and Céline M. F. Swennenhuis. Parameterized problems complete for nondeterministic FPT time and logarithmic space. In *Proceedings 62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021*, pages 193–204, 2021. doi:10.1109/FOCS52979.2021.00027.
5. Hans L. Bodlaender, Daniel Lokshtanov, and Eelko Penninkx. Planar capacitated dominating set is $W[1]$-hard. In Jianer Chen and Fedor V. Fomin, editors, *Proceedings 4th International Workshop on Parameterized and Exact Computation, IWPEC 2009*, volume 5917 of *Lecture Notes in Computer Science*, pages 50–60. Springer, 2009. doi:10.1007/978-3-642-11269-0_4.
6. Nick Brettell, Jake Horsfield, Andrea Munaro, and Daniël Paulusma. List k-colouring P_t-free graphs: A mim-width perspective. *Inf. Process. Lett.*, 173:106168, 2022. doi:10.1016/j.ipl.2021.106168.
7. Hajo Broersma, Petr A. Golovach, and Viresh Patel. Tight complexity bounds for FPT subgraph problems parameterized by the clique-width. *Theor. Comput. Sci.*, 485:69–84, 2013. doi:10.1016/j.tcs.2013.03.008.
8. Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems. *Theor. Comput. Sci.*, 511:66–76, 2013. doi:10.1016/j.tcs.2013.01.009.
9. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
10. Michael Dom, Daniel Lokshtanov, Saket Saurabh, and Yngve Villanger. Capacitated domination and covering: A parameterized perspective. In Martin Grohe and Rolf Niedermeier, editors, *Proceedings 3rd International Workshop on Parameterized and Exact Computation, IWPEC 2008*, volume 5018 of *Lecture Notes in Computer Science*, pages 78–90. Springer, 2008. doi:10.1007/978-3-540-79723-4_9.
11. Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Springer, 1999.

12 Michael Elberfeld, Christoph Stockhusen, and Till Tantau. On the space and circuit complexity of parameterized problems: Classes and completeness. *Algorithmica*, 71(3):661–701, 2015. doi:10.1007/s00453-014-9944-y.

13 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. Intractability of clique-width parameterizations. *SIAM J. Comput.*, 39(5):1941–1956, 2010. doi:10.1137/080742270.

14 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. Almost optimal lower bounds for problems parameterized by clique-width. *SIAM J. Comput.*, 43(5):1541–1563, 2014. doi:10.1137/130910932.

15 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Clique-width III: Hamiltonian cycle and the odd case of graph coloring. *ACM Trans. Algorithms*, 15(1):9:1–9:27, 2019. doi:10.1145/3280824.

16 Fedor V. Fomin, Petr A. Golovach, and Jean-Florent Raymond. On the tractability of optimization problems on H-graphs. *Algorithmica*, 82(9):2432–2473, 2020. doi:10.1007/s00453-020-00692-9.

17 Sylvain Guillemot. Parameterized complexity and approximability of the Longest Compatible Sequence problem. *Discret. Optim.*, 8(1):50–60, 2011. doi:10.1016/j.disopt.2010.08.003.

18 Frank Gurski and Egon Wanke. Line graphs of bounded clique-width. *Discret. Math.*, 307(22):2734–2754, 2007. doi:10.1016/j.disc.2007.01.020.

19 Lars Jaffke, O-joung Kwon, and Jan Arne Telle. Mim-width II. The feedback vertex set problem. *Algorithmica*, 82:118–145, 2020.

20 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, 2018. doi:10.1145/3170442.

21 Luke Mathieson and Stefan Szeider. The parameterized complexity of regular subgraph problems and generalizations. In James Harland and Prabhu Manyem, editors, *Proceedings 14th Computing: The Australasian Theory Symposium, CATS 2008*, volume 77 of *CRPIT*, pages 79–86. Australian Computer Society, 2008. URL: http://crpit.scem.westernsydney.edu.au/abstracts/CRPITV77Mathieson.html.

22 Hannes Moser and Dimitrios M. Thilikos. Parameterized complexity of finding regular induced subgraphs. *J. Discrete Algorithms*, 7(2):181–190, 2009. doi:10.1016/j.jda.2008.09.005.

23 Michal Pilipczuk and Marcin Wrochna. On space efficiency of algorithms working on structural decompositions of graphs. *ACM Trans. Comput. Theory*, 9(4):18:1–18:36, 2018. doi:10.1145/3154856.

24 Martin Vatshelle. *New Width Parameters of Graphs*. PhD thesis, University of Bergen, Norway, 2012.

25 Egon Wanke. k-NLC graphs and polynomial algorithms. *Discret. Appl. Math.*, 54(2-3):251–266, 1994. doi:10.1016/0166-218X(94)90026-4.