

Sparse Induced Subgraphs in P_7 -Free Graphs of Bounded Clique Number

Maria Chudnovsky  

Princeton University, NJ, USA

Jadwiga Czyżewska  

University of Warsaw, Poland

Kacper Kluk  

University of Warsaw, Poland

Marcin Pilipczuk  

University of Warsaw, Poland

Paweł Rzażewski  

Warsaw University of Technology, Poland

University of Warsaw, Poland

Abstract

Many natural computational problems, including e.g. MAX WEIGHT INDEPENDENT SET, FEEDBACK VERTEX SET, or VERTEX PLANARIZATION, can be unified under an umbrella of finding the largest sparse induced subgraph that satisfies some property definable in CMSO_2 logic. It is believed that each problem expressible with this formalism can be solved in polynomial time in graphs that exclude a fixed path as an induced subgraph. This belief is supported by the existence of a quasipolynomial-time algorithm by Gartland, Lokshtanov, Pilipczuk, Pilipczuk, and Rzażewski [STOC 2021], and a recent polynomial-time algorithm for P_6 -free graphs by Chudnovsky, McCarty, Pilipczuk, Pilipczuk, and Rzażewski [SODA 2024].

In this work we extend polynomial-time tractability of all such problems to P_7 -free graphs of bounded clique number.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases P_t -free graphs, maximum weight induced subgraph, maximum weight independent set

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2025.20

Related Version *Full Version*: <https://arxiv.org/abs/2412.14836>

Funding *Maria Chudnovsky*: Supported by NSF Grant DMS-2348219 and by AFOSR grant FA9550-22-1-0083.

Jadwiga Czyżewska: Supported by Polish National Science Centre SONATA BIS-12 grant number 2022/46/E/ST6/00143.

Kacper Kluk: Supported by Polish National Science Centre SONATA BIS-12 grant number 2022/46/E/ST6/00143.

Marcin Pilipczuk: Supported by Polish National Science Centre SONATA BIS-12 grant number 2022/46/E/ST6/00143.

Paweł Rzażewski: Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme grant agreement number 948057.



© Maria Chudnovsky, Jadwiga Czyżewska, Kacper Kluk, Marcin Pilipczuk, and Paweł Rzażewski; licensed under Creative Commons License CC-BY 4.0

36th International Symposium on Algorithms and Computation (ISAAC 2025).

Editors: Ho-Lin Chen, Wing-Kai Hon, and Meng-Tsung Tsai; Article No. 20; pp. 20:1–20:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

When studying computationally hard problems, a natural question to investigate is whether they become tractable when input instances are somehow “well-structured”. In particular, in algorithmic graph theory, we often study the complexity of certain (hard in general) problems, when restricted to specific graph classes. Significant attention is given to classes that are *hereditary*, i.e., closed under vertex deletion.

Potential maximal cliques. Arguably, the problem that is best studied in this context is MAX WEIGHT INDEPENDENT SET (MWIS) in which we are given a vertex-weighted graph and we ask for a set of pairwise non-adjacent vertices of maximum possible weight. Investigating the complexity of MWIS in restricted graph classes led to discovering numerous new tools and techniques in algorithmic graph theory [24, 21, 18, 16]. One of such general techniques is the framework of *potential maximal cliques* by Bouchitté and Todinca [7, 8]. Intuitively speaking, a potential maximal clique (or PMC for short) in a graph G is a bag of a “reasonable” tree decomposition of G . The key contribution of Bouchitté and Todinca [7, 8] is showing that:

1. the family $\mathcal{F}$ of PMCs in a graph G can be enumerated in time polynomial in $|V(G)|$ and $|\mathcal{F}|$, and
2. given a family $\mathcal{F}$ containing *all* PMCs of G (and possibly some other sets as well), MWIS can be solved in time polynomial in $|V(G)|$ and $|\mathcal{F}|$ by mimicking natural dynamic programming on an appropriate (but unknown) tree decomposition.

Consequently, MWIS admits a polynomial-time algorithm when restricted to any class with polynomially many PMCs. (Here we say that a class $\mathcal{X}$ of graphs has polynomially many PMCs if there exists a polynomial p , such that the number of PMCs in any n -vertex graph in $\mathcal{X}$ is at most $p(n)$.)

Later this framework was extended by Fomin, Todinca, and Villanger [19] to the problem of finding a large “sparse” (here meaning: of bounded treewidth) induced subgraph satisfying certain CMSO₂ formula ψ . (CMSO₂ stands for *Counting Monadic Second Order* logic, which is a logic where one can use vertex/edge variables, vertex/edge set variables, quantifications over these variables, and standard propositional operands.) Formally, for a given integer d and a fixed CMSO₂ formula ψ with one free set variable, the $(\text{tw} \leq d, \psi)$ -MWIS problem (here “MWIS” stands for “max weight induced subgraph”) is defined as follows.

$(\text{tw} \leq d, \psi)$ -MWIS

Input: A graph G equipped with a weight function $w: V(G) \rightarrow \mathbb{Q}_+$.

Task: Find a pair (Sol, X) such that

- $X \subseteq \text{Sol} \subseteq V(G)$,
 - $G[\text{Sol}]$ is of treewidth at most d ,
 - $G[\text{Sol}] \models \psi(X)$,
 - X is of maximum weight subject to the conditions above,
- or conclude that no such pair exists.

As shown by Fomin, Todinca, and Villanger [19], every problem expressible in this formalism can be solved in polynomial time on classes of graphs with polynomially many potential maximal cliques.

Note that if $d = 0$ and ψ is a formula satisfied by all sets, then $(\text{tw} \leq d, \psi)$ -MWIS is exactly MWIS. It turns out that $(\text{tw} \leq d, \psi)$ -MWIS captures several other well known computational problems (see also the discussion in [19, 29]), e.g.;

- FEEDBACK VERTEX SET (one of Karp’s original 21 NP-complete problems [27]), equivalent by complementation to finding an induced forest of maximum weight,
- EVEN CYCLE TRANSVERSAL [34, 33, 2], equivalent by complementation to finding an induced graph whose every block is an odd cycle,
- finding a largest weight induced subgraph whose every component is a cycle,
- finding a maximum number of pairwise disjoint and non-adjacent induced cycles.

MWIS in graphs excluding a long induced path. Despite all advantages of the Bouchitté-Todinca framework, its applicability is somehow limited, as there are numerous natural hereditary graph classes that *do not* have polynomially many PMCs. Can we use at least some parts of the framework outside its natural habitat?

A notorious open question in algorithmic graph theory is whether MWIS (and, more generally, $(\text{tw} \leq d, \psi)$ -MWIS) can be solved in polynomial time in graphs that exclude a fixed path as an induced subgraph. It is believed that this question has an affirmative answer, which is supported by the existence of a quasipolynomial-time algorithm of Gartland, Lokshtanov, Pilipczuk, Pilipczuk, and Rzażewski [20, 36, 22]. However, if it comes to polynomial-time algorithms, we know much less.

A polynomial-time algorithm for MWIS (as well as for many other problems) in P_4 -free graphs (i.e., graphs that exclude a 4-vertex path as an induced subgraph; analogously we define P_t -free graphs for any t), was discovered already in 1980s [14]. In today’s terms we would say that these graphs have bounded clique-width and thus polynomial-time algorithms for many natural problems, including $(\text{tw} \leq d, \psi)$ -MWIS, follow from a celebrated theorem by Courcelle, Makowsky, and Rotics [15]. However, already the P_5 -free case proved to be quite challenging. In 2014, Lokshtanov, Vatshelle, and Villanger [30] showed that MAX WEIGHT INDEPENDENT SET admits a polynomial-time algorithm in P_5 -free graphs – by adapting the framework of Bouchitté and Todinca. Let us emphasize that P_5 -free graphs might have exponentially many PMCs, so the approach discussed above cannot be applied directly. Instead, Lokshtanov, Vatshelle, and Villanger proved that in polynomial time one can enumerate a family of *some* PMCs, and this family is sufficient to solve MWIS via dynamic programming.

By extending this method (and adding a significant layer of technical complicacy), Grzesik, Klimošová, Pilipczuk, and Pilipczuk [23] managed to show that MWIS is polynomially-solvable in P_6 -free graphs.

Some time later, Abrishami, Chudnovsky, Pilipczuk, Rzażewski, and Seymour [1] revisited the P_5 -free case and introduced a major twist to the method: they proved that in order to solve MWIS and its generalizations, one does not have to have PMCs exactly, but it is sufficient to enumerate their *containers*: supersets that do not introduce any new vertices of the (unknown) optimum solution. They also proved that containers for *all* PMCs in P_5 -free graphs can be enumerated in polynomial time, circumventing the problem of exponentially many PMCs and significantly simplifying the approach of Lokshtanov, Vatshelle, and Villanger [30]. This result yields a polynomial-time algorithm for $(\text{tw} \leq d, \psi)$ -MWIS in P_5 -free graphs.

By further relaxing the notion of a container (to a *carver*), Chudnovsky, McCarty, Pilipczuk, Pilipczuk, and Rzażewski [13] managed to extend polynomial-time solvability of $(\text{tw} \leq d, \psi)$ -MWIS to the class of P_6 -free graphs.

Our contribution. In this work we push the boundary of tractability of $(\text{tw} \leq d, \psi)$ -MWIS in P_t -free graphs by showing the following result.

► **Theorem 1.1.** *For every fixed integers d, k , and a CMSO₂ formula ψ , the $(\text{tw} \leq d, \psi)$ -MWIS problem can be solved in polynomial time in P_7 -free graphs with clique number at most k .*

Quite interestingly, in P_t -free graphs, the boundedness of treewidth is equivalent to the boundedness of degeneracy and to the boundedness of treedepth [4]. Thus, Theorem 1.1 yields a polynomial-time algorithm for problems like:

- VERTEX PLANARIZATION [26, 35], which asks for a largest induced planar subgraph, or
- for constant k , finding a largest set of vertices inducing a subgraph of maximum degree at most k (see, e.g., [25]).

Let us remark that the idea of investigating P_t -free graphs of bounded clique number was considered before. Pilipczuk and Rzażewski [37] showed that P_6 -free graphs of bounded clique number have polynomially many PMCs and thus the framework of Bouchitté and Todinca can be applied here directly. However, this is no longer true for P_7 -free graphs (even bipartite) (see the example in the Conclusions in [13]). On the other hand, Brandstädt and Mosca [9] provided a polynomial-time algorithm for MWIS in P_7 -free triangle-free graphs. However, they used a rather ad-hoc approach that works well for MWIS, but gives little hope to generalize it to more complicated instances of $(\text{tw} \leq d, \psi)$ -MWIS. Our work vastly extends both these results.

2 Technical overview

For standard graph notation used in this work, including the definition of a tree decomposition, we refer to the preliminaries section of the full version in the appendix.

As mentioned, in P_t -free graphs the width parameters treewidth, treedepth, and degeneracy are functionally equivalent [4]. Furthermore, as discussed in [13], the property of having treewidth, treedepth, or degeneracy at most d can be expressed as a CMSO₂ formula of size depending on d only. Consequently, if we define problems $(\text{td} \leq d, \psi)$ -MWIS and $(\text{deg} \leq d, \psi)$ -MWIS analogously as $(\text{tw} \leq d, \psi)$ -MWIS, but with respect to treedepth and degeneracy, these three formalisms describe the same family of problems, when restricted to P_t -free graphs.

Treedepth structures. Following [13], the treedepth formalism is the most handy; we henceforth work with $(\text{td} \leq d, \psi)$ -MWIS. Furthermore, it is more convenient to work not just with induced subgraphs of bounded treedepth, but with *treedepth- d structures*: induced subgraphs of treedepth at most d with a fixed elimination forest. Let us proceed with formal definitions.

The *rooted forest* $\mathcal{T}$ is a forest in which each component has exactly one distinguished vertex, called a *root*. A path in a rooted forest is called *vertical* if it connects a vertex and any of its ancestors. Given a vertex v , we define its *depth* as the number of vertices on the path connecting v and the root (so the root has depth 1). The *height* of the rooted forest $\mathcal{T}$ is equal to the maximum depth of a vertex of $\mathcal{T}$. By $\mathcal{T}^\alpha$ we denote a set of vertices of $\mathcal{T}$ of depth exactly α . We also write $\mathcal{T}^{>\alpha}$ for $\bigcup_{\alpha' > \alpha} \mathcal{T}^{\alpha'}$.

We say that vertices u and v are $\mathcal{T}$ -*comparable* if they can be connected via a vertical path. Otherwise, we say that they are $\mathcal{T}$ -*incomparable*. An *elimination forest* of graph G is a rooted forest $\mathcal{T}$ such that $V(\mathcal{T}) = V(G)$ and for each edge $uv \in E(G)$ vertices u and v are $\mathcal{T}$ -comparable. The *treedepth* of graph G is the minimum height of any possible elimination forest of graph G .

Given a graph G and an integer d , a *treedepth- d structure* (the notion first proposed in [13]) is a rooted forest $\mathcal{T}$ of height at most d such that $V(\mathcal{T})$ is a subset of $V(G)$ and $\mathcal{T}$ is an elimination forest of the subgraph of G induced by $V(\mathcal{T})$. A treedepth- d structure $\mathcal{T}$ is a *substructure* of a treedepth- d structure $\mathcal{T}'$ if $\mathcal{T}$ is a subgraph of $\mathcal{T}'$ (as a rooted forest) and every root of $\mathcal{T}$ is also a root of $\mathcal{T}'$. A treedepth- d structure $\mathcal{T}$ is called *maximal* if there is no treedepth- d structure $\mathcal{T}'$ such that $\mathcal{T}$ is a substructure of $\mathcal{T}'$ and $\mathcal{T} \neq \mathcal{T}'$. Equivalently, $\mathcal{T}$ is maximal if it cannot be extended by adding a leaf preserving the bound on the height of $\mathcal{T}$. To avoid notational clutter, we sometimes treat $\mathcal{T}$ as set of vertices, for example in expressions $|A \cap \mathcal{T}|$.

Using treedepth structures. Let $\mathcal{T}$ be a treedepth- d structure in G ; think of $\mathcal{T}$ as the sought solution to the $(\text{td} \leq d, \psi)$ -MWIS problem in question. As proven in [19] (and cast onto the current setting in [13]), there exists a tree decomposition (T, β) of G whose every bag is either contained in $N[v]$ for a leaf v of $\mathcal{T}$, or whose intersection with $\mathcal{T}$ is contained in a single root-to-leaf path of $\mathcal{T}$, excluding the leaf. We call the latter bags *$\mathcal{T}$ -avoiding*.

Furthermore, the framework of [19] shows how to solve $(\text{td} \leq d, \psi)$ -MWIS given a family $\mathcal{F}$ of subsets of $V(G)$ with the promise that all bags of such tree decomposition (T, β) are in $\mathcal{F}$. In [1], it is shown that it suffices for $\mathcal{F}$ to contain only *containers of bounded defect* for bags in (T, β) : we require that there exists a universal constant $\hat{d}$ such that for every $t \in V(T)$ there exists $A \in \mathcal{F}$ such that the bag $\beta(t)$ is contained in A and A contains at most $\hat{d}$ elements of $\mathcal{T}$.

Note that if v is a leaf of $\mathcal{T}$, then $N[v]$ may contain only ancestors of v among the vertices of $\mathcal{T}$, so $N[v]$ is an excellent container for any bag contained in $N[v]$. Thus, it suffices to provide containers for $\mathcal{T}$ -avoiding bags. This is how [1] solved $(\text{td} \leq d, \psi)$ -MWIS in P_5 -free graphs; by generalizing the definition of a container to a carver, the same approach is applied to P_6 -free graphs in [13].

In fact, for a treedepth- d structure $\mathcal{T}$, the tree decomposition (T, β) of [19] is constructed as follows: it is shown that there exists a minimal chordal completion F of G (i.e., an inclusion-wise minimal set of edges to add to G to make it chordal) whose addition keeps $\mathcal{T}$ a treedepth- d structure, and (T, β) is any clique tree of $G + F$. A *potential maximal clique* (abbreviated as PMC) is a maximal clique in $G + F$ for *some* minimal chordal completion F of G . Both [1] and [13] actually provide containers/carvers for *every* treedepth- d structure in G and an *arbitrary* choice of a $\mathcal{T}$ -avoiding PMC in G . Furthermore, [13] provides an example of a family of P_7 -free triangle-free graphs where such a statement is impossible (i.e., any such family of containers/carvers would need to be of exponential size).

However, the algorithmic framework of [19, 1, 13] requires only that the provided family $\mathcal{F}$ contains containers/carvers for all bags of *only one* “good” tree decomposition (T, β) , and only for the sought solution $\mathcal{T}$. This should be contrasted with all $\mathcal{T}$ -avoiding PMCs, which is in some sense the set of all reasonable $\mathcal{T}$ -avoiding bags, and for all treedepth- d structures $\mathcal{T}$. Thus, if we want to tackle P_7 -free graphs of bounded clique number, we need to significantly deviate from the path of [1, 13] and either use the power of the choice of (T, β) or the fact that we need to handle only $\mathcal{T}$ being the optimum solution to the problem we are solving.

From PMCs to minimal separators. The assumption of bounded clique number allows us to quickly reduce the case of finding a container for a PMC to finding a container for a minimal separator. For a set $S \subseteq V(G)$, a connected component A of $G - S$ is *full to S* if

$N(A) = S$; a set S is a *minimal separator* if it has at least two full components.¹ A classic fact about PMCs [8] is that for every potential maximal clique Ω and for every connected component D of $G - \Omega$, the set $N(D)$ is a minimal separator. By a classic result of Gyárfás, the class of P_t -free graphs is χ -bounded, in particular, if G is P_t -free and $\omega(G) \leq k$ (where $\omega(G)$ is the number of vertices in a largest clique in G), then $\chi(G) \leq (t-1)^{k-1}$. This, combined with a VC-dimension argument based on the classic result of Ding, Seymour, and Winkler [17], shows the following:

► **Lemma 2.1.** *For every t and k there exists c such that for every P_t -free graph G with $\omega(G) \leq k$ and any PMC Ω of G one of the two following conditions holds:*

1. *there exists $v \in \Omega$ such that $\Omega = N[v]$, or*
2. *there exists a family $\mathcal{D} \subseteq \text{cc}(G - \Omega)$ of size at most c such that $\Omega = \bigcup_{D \in \mathcal{D}} N(D)$.*

That is, except for a simple case $N[v] = \Omega$ for some $v \in \Omega$, the PMC Ω is a union of a constant number of minimal separators. Hence, it suffices to generate a family $\mathcal{F}'$ of containers for $\mathcal{T}$ -avoiding minimal separators (which are defined analogously to $\mathcal{T}$ -avoiding bags) and take $\mathcal{F}$ to be the family of unions of all tuples of at most c elements of $\mathcal{F}'$.

Capturing minimal separators. Let G be a P_7 -free graph with $\omega(G) \leq k$. We start as in [1, 13]: let $\mathcal{T}$ be an arbitrary treedepth- d structure in G and let S be a minimal separator in G with full components A and B such that S is $\mathcal{T}$ -avoiding. We want to design a polynomial-time algorithm that outputs a family $\mathcal{F}'$ of subsets of $V(G)$ that contains a container for S . The algorithm naturally does not know $\mathcal{T}$ nor S ; the convenient and natural way of describing the algorithm as performing some nondeterministic guesses about $\mathcal{T}$ and S , with the goal of outputting a container for S in the end. We succeed if the number of subcases coming from the guesses is polynomial in the size of G and the family $\mathcal{F}'$ consists of all containers generated by all possible runs of the algorithm.

We almost succeed with this quest. That is, we are able to guess an “almost container” K for S : K contains a constant number of vertices of $\mathcal{T}$ and we identified a set $\mathcal{D}$ of *tricky* connected components of $G - K$ that are contained in $A \cup B \cup S$; all other connected components of $G - K$ are contained in A , in B , or in some other connected component of $G - S$. Every tricky component $D \in \mathcal{D}$ is somewhat simpler: we identify a subset $\emptyset \neq L(D) \subsetneq D$ such that if $\mathcal{C}(D)$ is the family of all connected components of $G[L(D)]$ and of $G[D \setminus L(D)]$, then every $C \in \mathcal{C}(D)$ is a module of $G[D]$. (A set $A \subseteq V(G)$ is a module in G if every vertex $v \in V(G) \setminus A$ is either complete to A or anti-complete to A .)

Observe now that every connected component $C \in \mathcal{C}(D)$ satisfies $\omega(G[C]) < \omega(G) \leq k$, as $D \setminus C$ contains a vertex complete to C . Hence, we can recurse on C , understanding it fully (more precisely, finding optimum partial solutions; the definition of this “partial” requires a lot of CMSO₂ mumbling).

Furthermore, we observe that the quotient graph $G[D]/\mathcal{C}(D)$ is bipartite. Thus, if we can understand how to solve $(\text{td} \leq d, \psi)$ -MWIS on P_7 -free *bipartite* graphs, then we should be done: the partial solutions from components $C \in \mathcal{C}(D)$, combined with the understanding of the P_7 -free bipartite graph $G[D]/\mathcal{C}(D)$ should give all partial solutions of $G[D]$ for every $D \in \mathcal{D}$. This, in turn, should give an understanding on how $(\text{td} \leq d, \psi)$ -MWIS behaves on $K \cup \bigcup \mathcal{D}$. As $S \subseteq K \cup \bigcup \mathcal{D}$, this should suffice to solve $(\text{td} \leq d, \psi)$ -MWIS on G using $(K, \mathcal{D}, L)$ to play the role of the container for S .

¹ Note the following equivalent definition: We say that a set of vertices S is a (u, v) -separator if u and v belong to distinct components in $G - S$. A (u, v) -separator S is called *minimal* if no proper subset of S is a (u, v) -separator. A set of vertices S is called a *minimal separator* of graph G if there exists a pair of vertices u and v such that S is a minimal (u, v) -separator.

The bipartite case. Explaining properly all “should”s from the previous paragraph is quite involved and tedious. In this overview, let us focus on the more interesting case of solving $(\text{td} \leq d, \psi)$ -MWIS in P_7 -free bipartite graphs. Here, we actually prove the following result.

► **Theorem 2.2.** *There exists an algorithm that, given a P_7 -free bipartite graph G and an integer $d > 0$, runs in time $n^{2^{2^{\mathcal{O}(d^3)}}}$ and computes a family $\mathcal{C}$ of subsets of $V(G)$ with the following guarantee: for every $\text{Sol} \subseteq V(G)$ such that $G[\text{Sol}]$ is of treedepth at most d , there exists a tree decomposition (T, β) of G such that for every $t \in V(T)$ we have $\beta(t) \in \mathcal{C}$ and $|\beta(t) \cap \text{Sol}| = 2^{2^{\mathcal{O}(d^3)}}$.*

We remark that Theorem 2.2 is not needed if one just wants to solve the MWIS problem, as this problem can be solved in general bipartite graphs using matching or flow techniques.

On the other hand, our work identified the bipartite case as an interesting subcase in exploring tractability of $(\text{td} \leq d, \psi)$ -MWIS in P_t -free graphs. To state a precise problem for future work, we propose polynomial-time tractability of FEEDBACK VERTEX SET in P_t -free bipartite graphs. Meanwhile, Theorem 2.2 and its proof in Section 3 can be of independent interest.

The proof of Theorem 2.2 starts from the work of Kloks, Liu, and Poon [28], who showed that *chordal bipartite* graphs have polynomial number of PMCs, and thus $(\text{td} \leq d, \psi)$ -MWIS problem is solvable in this graph class by the direct application of the PMC framework of Bouchitté and Todinca. (A graph G is *chordal bipartite* if it is bipartite and does not contain induced cycles longer than 4.) A P_7 -free bipartite graph is *almost* chordal bipartite: it can contain six-vertex cycles.

We would like to add some edges to the input graph G so that it becomes chordal bipartite. Let $C = c_1 - c_2 - \dots - c_6 - c_1$ be an induced six-vertex cycle in G ; we would like to add the edge c_1c_4 to keep G bipartite but break C . We show that, if one chooses C carefully, one can do it so that G remains P_7 -free. However, such an addition may break the sought solution $\mathcal{T}$: if $c_1, c_4 \in V(\mathcal{T})$ but are incomparable in the elimination forest $\mathcal{T}$, then $\mathcal{T}$ is no longer a treedepth- d structure in $G + \{c_1c_4\}$.

We remedy this by a thorough investigation of the structure of the neighborhood of a six-vertex cycle in a P_7 -free graph, loosely inspired by [5]. On a high level, we show that there is a branching process with a polynomial number of outcomes that, in some sense “correctly” completes G to a chordal bipartite graph, proving Theorem 2.2.

In summary, the proof of Theorem 1.1 consists of three ingredients. First, we show the guesswork that leads to a polynomial number of candidate “almost containers” $(K, \mathcal{D}, L)$ for a minimal separator S in the input graph G . Second, we focus on bipartite graphs and prove Theorem 2.2. Finally, we show how these two tools combine with the dynamic programming framework of [19, 1, 13].

In this extended abstract, we present a more detailed overview of the proof of one part, namely Theorem 2.2. The full version of the work can be found in the appendix.

3 P_7 -free bipartite graphs: Overview of the proof of Theorem 2.2

In this section we focus on P_7 -free bipartite graphs. We prove the following theorem.

► **Theorem 2.2.** *There exists an algorithm that, given a P_7 -free bipartite graph G and an integer $d > 0$, runs in time $n^{2^{2^{\mathcal{O}(d^3)}}}$ and computes a family $\mathcal{C}$ of subsets of $V(G)$ with the following guarantee: for every $\text{Sol} \subseteq V(G)$ such that $G[\text{Sol}]$ is of treedepth at most d , there exists a tree decomposition (T, β) of G such that for every $t \in V(T)$ we have $\beta(t) \in \mathcal{C}$ and $|\beta(t) \cap \text{Sol}| = 2^{2^{\mathcal{O}(d^3)}}$.*

The general approach is to reduce to the case of *chordal bipartite graphs*. Recall that a bipartite graph is *chordal bipartite* if its every cycle of length longer than 4 has a chord (i.e., the only induced cycles are of length 4).

The class of MWIS problems is tractable on chordal bipartite graphs thanks to the following result of Kloks, Liu, and Poon (and the general algorithm of Fomin, Todinca, and Villanger [19]).

► **Theorem 3.1** (Corollary 2 of [28]). *A chordal bipartite graph on n vertices and m edges has $\mathcal{O}(n + m)$ minimal separators.*

Observe that P_7 -free bipartite graphs are “almost” chordal bipartite: they only additionally allow C_6 as an induced subgraph. Our approach is to add edges to the input graph so that it becomes chordal bipartite, without destroying the sought solution. To this end, the following folklore characterization will be handy. (We use ♠ to mark statements whose proofs are omitted in this extended abstract due to space constraints.)

► **Lemma 3.2** (folklore, ♠). *Let G be a bipartite graph with bipartition V_1, V_2 . Then, G is chordal bipartite if and only if for every minimal separator S of G , $S \cap V_1$ is complete to $S \cap V_2$ (i.e., the separator induces a complete bipartite graph, also called a biclique).*

Lemma 3.2 motivates the following process of completing a P_7 -free bipartite graph G into a chordal bipartite graph: while G contains a minimal separator S that violates the statement of Lemma 3.2, complete $G[S]$ into a biclique. In Section 3.1 we analyse this process and show that it is well-behaved on P_7 -free graphs: it does not lead outside the class of P_7 -free bipartite graphs.

3.1 Completing to a chordal bipartite graph

Throughout the rest of this section we assume that the input graph G has a fixed bipartition into sets V_1, V_2 (this is an ordered partition). We remark that we will often look at certain induced subgraphs of G that are not necessarily connected, but a vertex never changes its side of the bipartition.

Lemma 3.2 motivates the following definition. Let G be a bipartite graph with bipartition V_1, V_2 , and let S be a minimal separator of G . We say that S *induces a biclique* if $S \cap V_1$ is complete to $S \cap V_2$. The operation of *completing S into a biclique* turns G into a graph $G + F := (V(G), E(G) \cup F)$, where $F = \{uv \mid u \in S \cap V_1, v \in S \cap V_2, uv \notin E(G)\}$.

The next two lemmata (proved by a tedious case analysis) are pivotal to our completion process.

► **Lemma 3.3** (♠). *Let G be a P_7 -free bipartite graph. Let S be a minimal separator. Let $G + F$ be the result of completing S into a biclique. Then $G + F$ is also P_7 -free.*

► **Lemma 3.4** (♠). *Let G be a P_7 -free bipartite graph. Let S be a minimal separator in G . Let $G + F$ be the result of completing S into a biclique. Let C be an induced C_6 in $G + F$. Then, C is also an induced C_6 in G .*

Let G be a P_7 -free bipartite graph with fixed bipartition V_1, V_2 , and let $\mathcal{T}$ be a treedepth- d structure in G . A tuple (C, x, y) is a *bad C_6* (with respect to $\mathcal{T}$) in G if C is an induced six-vertex cycle in G , and $x \in V_1, y \in V_2$ are two vertices that are at the same time (a) opposite vertices of C , and (b) incomparable vertices of $\mathcal{T}$. That is, if one adds the edge xy to G (which can be done without violating the bipartiteness of G), then one breaks the treedepth- d structure $\mathcal{T}$, i.e., $\mathcal{T}$ is not a treedepth- d structure in $G + xy$.

We have the following simple corollary of Lemma 3.4.

► **Lemma 3.5.** *Let G be a P_7 -free bipartite graph, let $\mathcal{T}$ be a treedepth- d structure in G , let S be a minimal separator in G , and let $G + F$ be a result of completing S into a biclique. Assume that there is no bad C_6 in G with respect to $\mathcal{T}$. Then, $\mathcal{T}$ is a treedepth- d structure in $G + F$ and, furthermore, there is no bad C_6 in $G + F$ with respect to $\mathcal{T}$.*

Proof. By contradiction, suppose that $E(C) \cap F \neq \emptyset$. As S is in a biclique in $G + F$, $V(C) \cap S$ contains either two or three consecutive vertices of C . Thus, $P := C \setminus S$ is a path and belongs to exactly one component of $G - S$. Without loss of generality, we can assume that P does not lie in B , i.e., $C \cap B = \emptyset$.

Let $x, z \in V(C) \cap S$ be the two vertices of C adjacent on C to the vertices of P . Note that $xz \notin E(G)$: either $|V(C) \cap S| = 3$ and x, z are on the same bipartiteness side of G or $|V(C) \cap S| = 2$ and then $xz \in F$.

Let R be the shortest path from x to z via B . Then, $R \cup P$ induce a hole C' in G . Since R is of length at least 2, C' is not shorter than C . Since G is P_7 -free, C' is a six-vertex hole. This can only happen if R is of length 2 and $V(C) \cap S = \{x, y, z\}$ for some $y \in S$. Then, $C = x - y - z - r - q - p - x$, where p, q, r lie in a single component of $G - S$ different than B . Without loss of generality, we can assume that $yz \in F$. Then we can find a shortest path R' , which connects y and z via B ; it is of length at least 3. Then the path $R - r - q - p$ is an induced path in G on at least 7 vertices, a contradiction. ◀

We conclude this section with the following enumeration.

► **Lemma 3.6.** *There exists an algorithm that, given a P_7 -free bipartite graph G and an integer d , runs in polynomial time and returns a family $\mathcal{C}$ of subsets of $V(G)$ of size $\mathcal{O}(|V(G)|^5)$ with the following property: for every treedepth- d structure $\mathcal{T}$ in G that admits no bad C_6 , there exists a tree decomposition (T, β) of G such that for every $t \in V(T)$, the set $\beta(t)$ belongs to $\mathcal{C}$ and $\beta(t)$ contains at most d elements of $\mathcal{T}$.*

Proof. Consider the following process. Start with $\hat{G} := G$. While $\hat{G}$ is not chordal bipartite, find an induced six-vertex cycle C in G , fix two opposite vertices x and y in C , find a minimal separator S in G that contains x and y (note that any minimal separator separating the two components of $C - \{x, y\}$ would do), complete S into a biclique obtaining $\hat{G} + F$, and set $\hat{G} := \hat{G} + F$.

Lemma 3.3 ensures that $\hat{G}$ stays P_7 -free bipartite. Lemma 3.5 ensures that every treedepth- d structure $\mathcal{T}$ in G that admits no bad C_6 remains so in $\hat{G}$.

By Theorem 3.1, $\hat{G}$ has $\mathcal{O}(|V(G)|^2)$ minimal separators. Using [7, 8], this allows us to bound the number of potential maximal cliques of $\hat{G}$ by $\mathcal{O}(|V(G)|^5)$ and enumerate them in a family $\mathcal{C}$.

As shown in [13], there exists a minimal chordal completion F of $\hat{G}$ such that for every maximal clique Ω of $\hat{G} + F$, the set $\Omega \cap \mathcal{T}$ is contained in a single leaf-to-root path in $\mathcal{T}$ and thus is of size at most d . Since all these maximal cliques are enumerated in $\mathcal{C}$, any clique tree of $\hat{G} + F$ serves as the promised tree decomposition (T, β) . ◀

3.2 Cleaning

In this section we define a branching step that cleans a specific part of the graph. The step will be general enough to be applicable in many contexts.

Let G be a P_7 -free bipartite graph. We say that a triple (A, B, C) of pairwise disjoint vertex sets of G is $\mathbb{W}$ -free if there are no two anticomplete P_3 s of the form $A - B - C$. Note that such a configuration naturally appears in a P_7 -free bipartite graph if $A \cup C$ is in one side on the bipartition, B is in the other side, and there is an additional vertex v in the same side as B such that $A \subseteq N(v)$ but $C \cap N(v) = \emptyset$. In our case, $\mathbb{W}$ -free sets appear naturally in the following context.

► **Lemma 3.7.** *Let G be a P_7 -free bipartite graph with a fixed bipartition V_1, V_2 . Let A and C be two disjoint subsets contained in V_i , for some $i \in \{1, 2\}$, and let $B \subseteq V_{3-i}$. Furthermore, assume that there exists a vertex $v \in V_{3-i} \setminus B$ such that $A \subseteq N(v)$ but $C \cap N(v) = \emptyset$. Then, (A, B, C) is $\mathbb{W}$ -free.*

Proof. Any two anticomplete P_3 s of the form $A - B - C$, together with v , induce a P_7 in G . ◀

Recall that in our setting, we have some unknown treedepth- d structure $\mathcal{T}$ in G and we are building a container for it. That is, we are happy with inserting any number of vertices of G into the container, as long as we are guaranteed that we insert only a constant number of vertices of $\mathcal{T}$ along the way. In the case of a $\mathbb{W}$ -free triple (A, B, C) , we would like to simplify this part of G by filtering out vertices of B that have neighbors both in A and in C . To achieve this goal, we will guess a set $X \subseteq A \cup B \cup C$ that on one hand contains (in one of the branches) at most a constant number of vertices of the fixed unknown $\mathcal{T}$, and on the other hand satisfies the following: no $b \in B \setminus X$ has a neighbor both in $A \setminus X$ and in $C \setminus X$.

To this end, we will rely on the following simple yet powerful observation. Observe that if (A, B, C) is $\mathbb{W}$ -free and both $A \cup C$ and B are independent sets (which happen, in particular, if $A \cup C$ is on one side of the bipartition and B is on the other side of the bipartition), then, for every distinct $b_1, b_2 \in B$ either $N(b_1) \cap A$ and $N(b_2) \cap A$ are comparable by inclusion or $N(b_1) \cap C$ and $N(b_2) \cap C$ are comparable by inclusion. This allows to use the following lemma on subsets of B , with the orders $\leq_1$ and $\leq_2$ being inclusion of the neighborhoods in A and C , respectively.

► **Lemma 3.8** (see Lemma 4.1 in [23]). *Let $(X, \leq_1)$ and $(X, \leq_2)$ be two partial orders on the set X such that any pair of elements of X is comparable in $\leq_1$ or in $\leq_2$. Then, there exists an element v such that for any element $x \in X$ we have $v \leq_1 x$ or $v \leq_2 x$.*

This observation is the engine of the following lemma that formalizes our goal of filtering out vertices of B that have neighbors both in A and in C .

► **Lemma 3.9.** *Fix an integer d . Let G be a graph, and let (A, B, C) be a $\mathbb{W}$ -free triple in G with both $A \cup C$ and B being independent set. Then one can in polynomial time enumerate a family $\mathcal{F}$ of subsets of $A \cup B \cup C$ of size at most $n^{4(d+1)^2}$ with the following guarantee:*

- *For every $X \in \mathcal{F}$, there is no $b \in B \setminus X$ with both $N(b) \cap (A \setminus X)$ and $N(b) \cap (C \setminus X)$ nonempty.*
- *For every treedepth- d structure $\mathcal{T}$ in G , there exists $X \in \mathcal{F}$ such that $|X \cap \mathcal{T}| \leq d^2(d+1)$.*

Proof. Fix a treedepth- d structure $\mathcal{T}$ in G . We will describe the process of enumerating the elements of $\mathcal{F}$ as a branching algorithm, guessing some properties of $\mathcal{T}$. The number of leaves of the branching will be polynomial in the size of G and in each leaf of the branching we will output one set X that satisfies the second property for every $\mathcal{T}$ that agrees with the guesses made in this leaf.

For every $b \in B$, let $\iota_A(b)$ be the maximum integer $1 \leq \alpha \leq d$ such that $N(b) \cap A \cap \mathcal{T}^\alpha$ contains at least two vertices; $\iota_A(b) = 0$ if such a α does not exist. Similarly define $\iota_C(b)$ with respect to $N(b) \cap C \cap \mathcal{T}^\alpha$. For $0 \leq \alpha, \beta \leq d$, let $B_{\alpha, \beta} = \{b \in B \mid \iota_A(b) = \alpha \wedge \iota_C(b) = \beta\}$. Note that sets $B_{\alpha, \beta}$ form a partition of B .

Initialize $X = \emptyset$. For every $0 \leq \alpha, \beta \leq d$, we will guess some set of vertices and include them into X . We will argue that there will be a branch where the vertices guessed for α, β ensure that every vertex from $B_{\alpha, \beta} \setminus X$ satisfies the first statement of the lemma. Thus, for (at least) one of sets X generated in the process, the statement will hold for every vertex in $B \setminus X$.

Consider fixed $0 \leq \alpha, \beta \leq d$. If $B_{\alpha, \beta}$ is empty, there is nothing to do, so assume otherwise. Define the following two orders $\leq_1$ and $\leq_2$ on $B_{\alpha, \beta}$:

$$\begin{aligned} b \leq_1 b' &\iff N(b) \cap A \supseteq N(b') \cap A, & \text{if } \alpha = 0, \\ b \leq_1 b' &\iff N(b) \cap A \cap \mathcal{T}^\alpha \subseteq N(b') \cap A \cap \mathcal{T}^\alpha & \text{if } \alpha > 0, \\ b \leq_2 b' &\iff N(b) \cap C \supseteq N(b') \cap C, & \text{if } \beta = 0, \\ b \leq_2 b' &\iff N(b) \cap C \cap \mathcal{T}^\beta \subseteq N(b') \cap C \cap \mathcal{T}^\beta & \text{if } \beta > 0. \end{aligned}$$

Apply Lemma 3.8 to $B_{\alpha, \beta}$ with $\leq_1$ and $\leq_2$, obtaining a vertex $b_{\alpha, \beta}^*$.

If $\alpha = 0$, guess $b_{\alpha, \beta}^*$ and add $N(b_{\alpha, \beta}^*) \cap A$ to X . This adds at most d vertices of $\mathcal{T}$ to X , while adds to X all vertices of $N(b) \cap A$ such that $b \in B_{\alpha, \beta}$ and $b_{\alpha, \beta}^* \leq_1 b$.

If $\alpha > 0$, guess two elements $a_{\alpha, \beta}^1, a_{\alpha, \beta}^2 \in N(b_{\alpha, \beta}^*) \cap A \cap \mathcal{T}^\alpha$ and add $N(a_{\alpha, \beta}^1) \cap N(a_{\alpha, \beta}^2) \cap B$ to X . This adds at most $(\alpha - 1)$ vertices of $\mathcal{T}$ to X , while adds to X every $b \in B_{\alpha, \beta}$ such that $b_{\alpha, \beta}^* \leq_1 b$.

Perform symmetrical operation for β and C : If $\beta = 0$, guess $b_{\alpha, \beta}^*$ and add $N(b_{\alpha, \beta}^*) \cap C$ to X . This adds at most d vertices of $\mathcal{T}$ to X , while adds to X all vertices of $N(b) \cap C$ such that $b \in B_{\alpha, \beta}$ and $b_{\alpha, \beta}^* \leq_2 b$. If $\beta > 0$, guess two elements $c_{\alpha, \beta}^1, c_{\alpha, \beta}^2 \in N(b_{\alpha, \beta}^*) \cap C \cap \mathcal{T}^\beta$ and add $N(c_{\alpha, \beta}^1) \cap N(c_{\alpha, \beta}^2) \cap B$ to X . This adds at most $(\beta - 1)$ vertices of $\mathcal{T}$ to X , while adds to X every $b \in B_{\alpha, \beta}$ such that $b_{\alpha, \beta}^* \leq_2 b$.

Finally, add the resulting set X to $\mathcal{F}$ if it satisfies the first bullet point of the statement.

We have already argued that in the branch when all guesses concerning $\mathcal{T}$ are correct, the first bullet point of the statement is satisfied. Furthermore, the size of $\mathcal{T} \cap X$ is bounded by

$$\sum_{\alpha=1}^d \sum_{\beta=1}^d (\alpha + \beta) = 2d \frac{d(d+1)}{2} = d^2(d+1).$$

Finally, observe that the number of branches is bounded by $n^{4(d+1)^2}$ as for every $0 \leq \alpha, \beta \leq d$ we guess at most four vertices of G . ◀

Thanks to Lemma 3.7, Lemma 3.9 is applicable in the following setting.

► **Definition 3.10.** Let G be a P_7 -free bipartite graph with a fixed bipartition V_1, V_2 and let $Z \subseteq V(G)$. The neighborhood partition with respect to Z is a partition $\mathcal{Z}$ of $V(G) \setminus Z$ into sets depending on (1) their side in the bipartition, and (2) their neighborhood in Z . More precisely,

$$\mathcal{Z} = \{A_{i,Y} \mid i \in \{1, 2\}, Y \subseteq Z \cap V_{3-i}\}, \quad \text{where } A_{i,Y} = \{v \in (V(G) \setminus Z) \cap V_i \mid N(v) \cap Z = Y\}.$$

► **Lemma 3.11.** Let G be a P_7 -free bipartite graph with a fixed bipartition V_1, V_2 , let $Z \subseteq V(G)$, and let $\mathcal{Z}$ be the neighborhood partition with respect to Z . Then, for every distinct $A, B, C \in \mathcal{Z}$, the triple (A, B, C) is $\mathbb{W}$ -free.

Proof. There is no P_3 of the form $A - B - C$ unless A and C are on one side of the bipartition, and B is on the other side of the bipartition. Then, by the definition of $\mathcal{Z}$, the vertices of A and the vertices of C differ in their neighborhood in Z : there exists $v \in Z$ such that either $N(v) \cap (A \cup C) = A$ or $N(v) \cap (A \cup C) = C$. Then, the claim follows from Lemma 3.7 and the symmetry of A, C . ◀

We summarize this section with the following cleaning step.

► **Lemma 3.12.** *Let G be a P_7 -free bipartite graph with a fixed bipartition V_1, V_2 , let $Z \subseteq V(G)$, and let $\mathcal{Z}$ be the neighborhood partition with respect to Z . Then one can in time $|V(G)|^{2^{3|Z|} \cdot \mathcal{O}(d^2)}$ enumerate a family $\mathcal{F}$ of at most $n^{4(d+1)^2 2^{3|Z|}}$ subsets of $V(G) \setminus Z$ with the following properties:*

- *For every $X \in \mathcal{F}$, for every $v \in V(G) \setminus X$, the elements of $N(v) \setminus (X \cup Z)$ are contained in a single set of $\mathcal{Z}$.*
- *For every treedepth- d structure $\mathcal{T}$ in G , there exists $X \in \mathcal{F}$ such that $|X \cap \mathcal{T}| \leq 2^{3|Z|} \cdot d^2(d+1)$.*

Proof. Observe that $|\mathcal{Z}| \leq 2^{|Z|} + 1$. Let $\mathcal{Z}^3$ be the family of all triples (A, B, C) where A, B, C are distinct elements of $\mathcal{Z}$. For every triple $(A, B, C) \in \mathcal{Z}^3$, we apply Lemma 3.9 obtaining a family $\mathcal{F}_{(A,B,C)}$; Lemma 3.11 asserts that the assumptions are satisfied. For every triple $(X_{(A,B,C)})_{(A,B,C) \in \mathcal{Z}^3} \in \prod_{(A,B,C) \in \mathcal{Z}^3} \mathcal{F}_{(A,B,C)}$ insert $\bigcup_{(A,B,C) \in \mathcal{Z}^3} X_{(A,B,C)}$ into $\mathcal{F}$. The promised guarantees and size bounds are immediate from Lemma 3.9 and the bound $|\mathcal{Z}| \leq 2^{|Z|} + 1$, which implies $|\mathcal{Z}^3| \leq (2^{|Z|} + 1) \cdot 2^{|Z|} \cdot (2^{|Z|} - 1) \leq 2^{3|Z|}$. ◀

3.3 Branching on a bad C_6 s

The tools developed in Section 3.1 allow us to solve $(td \leq d, \phi)$ -MWIS on G , assuming that there is no bad C_6 w.r.t. the solution treedepth- d structure $\mathcal{T}$. We now investigate properties of bad C_6 s and how to branch on them.

We will often denote an induced C_6 as $C = c_1 - c_2 - \dots - c_6 - c_1$. Then (c_1, c_4) , (c_2, c_5) , (c_3, c_6) are pairs of the opposite vertices in C . We implicitly assume that the indices behave cyclically, i.e., $c_7 = c_1$ etc. We will need the following observation implicit in [6].

► **Lemma 3.13 (♠).** *Let G be a P_7 -free bipartite graph, let $C = c_1 - c_2 - \dots - c_6 - c_1$ be an induced C_6 in G , and let D be a connected component of $G - N[V(C)]$ that contains at least two vertices. Then, for every $v \in N(D)$, the set $N(v) \cap V(C)$ equals to either $\{c_1, c_3, c_5\}$ or $\{c_2, c_4, c_6\}$.*

For an induced six-vertex-cycle $C = c_1 - c_2 - \dots - c_6 - c_1$, denote

$$\begin{aligned} S_1^C &= \{v \in N(V(C)) \mid N(v) \cap V(C) = \{c_1, c_3, c_5\}\}, \\ S_2^C &= \{v \in N(V(C)) \mid N(v) \cap V(C) = \{c_2, c_4, c_6\}\}. \end{aligned}$$

Furthermore, let MR_C be the union of vertex sets of all connected components of $G - N[V(C)]$ that contain at least 2 vertices. (MR stands for the “main remainder”.) With this notation, Lemma 3.13 states that $N(\text{MR}_C) \subseteq S_1^C \cup S_2^C$.

Greatly simplifying, our algorithm will guess a bad C_6 (C, x, y) , resolve $N[V(C)]$ using cleaning, and recurse on connected components of MR_C . To restrict the space of possible recursive calls, we need the following observation.

► **Lemma 3.14.** *Let G be a P_7 -free bipartite graph with a fixed bipartition V_1, V_2 , and let $\mathcal{B}$ be a family of pairwise disjoint and anticomplete subsets of $V(G)$ such that for every $B \in \mathcal{B}$, $G[B]$ is connected and $|B| > 1$. Let D be a connected component of $G - \bigcup_{B \in \mathcal{B}} N[B]$, let C be the connected component of G that contains D . Then, for every $i \in \{1, 2\}$, either $N(D) \cap V_i = \emptyset$ or there exists a single set $B_i \in \mathcal{B}$ such that $N(D) \cap V_i \subseteq N(B_i)$. Consequently, there exists a subfamily $\mathcal{B}' \subseteq \mathcal{B}$ of size at most 2 such that $N(D) \subseteq \bigcup_{B \in \mathcal{B}'} N[B]$ (i.e., D is a connected component of $G - \bigcup_{B \in \mathcal{B}'} N[B]$).*

Proof. Let $\mathcal{B}_i \subseteq \mathcal{B}$ be an inclusion-wise minimal set such that $\bigcup_{B \in \mathcal{B}_i} N(B) \supseteq N(D) \cap V_i$. Assume $|\mathcal{B}_i| > 1$; let $B, B' \in \mathcal{B}_i$ be distinct. By minimality, there exists $v, v' \in N(D) \cap V_i$ such that $v \in N(B) \setminus N(B')$ and $v' \in N(B') \setminus N(B)$. Since $|B|, |B'| > 1$, there exists an induced P_3 of the form $v - B - B$ and an induced P_3 of the form $v' - B' - B'$. Let Q be a shortest path from v to v' via D . Then, there exists an induced path in G of the form $B - B - v - Q - v' - B' - B'$ and this path has at least 7 vertices, a contradiction. $\blacktriangleleft$

To see how Lemma 3.14 is useful, consider a hypothetical recursive branching algorithm that guesses a bad C_6 (C, x, y) and recurses on connected components of MR_C . Assume that some recursive call is invoked on an induced subgraph $G[A]$ and the C_6 s in the stack of the recursion are $C^1, C^2, \dots, C^p$. Then, $\mathcal{B} = \{V(C^i) \mid 1 \leq i \leq p\}$ satisfies the prerequisites of Lemma 3.14 and $G[A]$ is a connected component of $G - \bigcup_{B \in \mathcal{B}} N[B]$. Lemma 3.14 asserts that there are at most two C_6 s among $C^1, C^2, \dots, C^p$, say C^a and C^b , such that $G[A]$ is a connected component of $G - (N[V(C^a)] \cup N[V(C^b)])$. Since $|V(C^a) \cup V(C^b)| = 12$, there are $\mathcal{O}(|V(G)|^{13})$ options for the set A , and we have a polynomial bound on the number of different possible recursive calls.

However, cleaning $N[V(C)]$ for a bad C_6 (C, x, y) is far from trivial, and we help ourselves with a careful choice of (C, x, y) . Informally speaking, we choose the one in which the depths of x and y in $\mathcal{T}$ are maximum possible. This maximality allows us to argue that some other C_6 s found in the vicinity of C are not bad, as their vertices of $\mathcal{T}$ are deeper.

Further complications arise from the fact that either S_1^C or S_2^C can contain an arbitrary number of vertices of $\mathcal{T}$ (but we can prove that one of these sets contains at most d vertices of $\mathcal{T}$). In this case, the following observation helps us get enough structure to perform branching.

Lemma 3.15. *Let G be a P_7 -free bipartite graph with a fixed bipartition V_1, V_2 , let $Z \subseteq V(G)$ be connected, and let D, D' be two connected components of $G - N[Z]$ that are of size at least 2 each. Then, for every $i \in \{1, 2\}$, the sets $N(D) \cap V_i$ and $N(D') \cap V_i$ are comparable by inclusion.*

Proof. Assume the contrary, let $x \in (N(D) \setminus N(D')) \cap V_i$ and $x' \in (N(D') \setminus N(D)) \cap V_i$. By the existence of x and x' , we observe that Z, x, x', D, D' lie all in the same connected component of G and $x, x' \in N(Z)$. Let Q be the shortest path from x to x' via Z . Since $|D|, |D'| > 1$, there exists an induced P_3 of the form $x - D - D$ and an induced P_3 of the form $x' - D' - D'$. But then there exists an induced path of the form $D - D - x - Q - x' - D' - D'$, which has at least 7 vertices, a contradiction. $\blacktriangleleft$

The most frequent usage of Lemma 3.15 will be when $Z = V(C)$ for some bad C_6 (C, x, y) . Then, D, D' are connected components of MR_C and Lemma 3.15 asserts that for every $i \in \{1, 2\}$, the neighborhoods $N(D) \cap S_i^C$ and $N(D') \cap S_i^C$ are comparable by inclusion.

This concludes the main insights into the proof of Theorem 2.2. Full proof can be found in the full version of the paper in the appendix.

4 Conclusion

Our work suggests some directions for future research. Of course the most ambitious goal is to show a polynomial-time algorithm for $(\text{tw} \leq d, \psi)$ -MWIS in P_t -free graphs, for all fixed t , with the first open case for $t = 7$ (with no further assumptions on the graph). However, there are several intermediate goals that also seem interesting. For example, we think that

the idea of considering graphs of bounded clique number is quite promising. Not only it allows us to use some strong structural tools like χ -boundedness or VC-dimension, but also to measure the progress of an algorithm by decreasing clique number (see also [12, 11]).

► **Problem 4.1** ([12, Problem 9.2]). *Show that for every fixed t and k , MWIS is polynomial-time solvable in P_t -free graphs with clique number at most k .*

As illustrated in Section 3, assuming additionally that a graph is bipartite might lead to an easier, but still interesting problem. Of course in this setting asking for an algorithm for MWIS makes little sense, but already the next case is far from trivial.

► **Problem 4.2.** *Show that for every fixed t , given a vertex-weighted bipartite P_t -free graph, in polynomial-time we can find an induced forest of maximum possible weight.*

Let us point out that in case $t = 7$ one can use an approach alternative to our Theorem 2.2. Indeed, it follows from the work of Lozin and Zamaraev [32] that P_7 -free bipartite graphs have bounded mim-width, which allows us to solve maximum induced forest in polynomial time [3]. However, already P_8 -free bipartite graphs have unbounded mim-width [10].

A different subclass of P_t -free graphs are $(P_t, K_{1,k})$ -free graphs (i.e., excluding additionally a star with k leaves as an induced subgraph), studied by Lozin and Rautenbach [31], who show that MAXIMUM WEIGHT INDEPENDENT SET is polynomial-time solvable in these graph classes. Does this result generalize to $(\text{tw} \leq d, \psi)$ -MWIS?

Finally, let us point out that Abrishami, Chudnovsky, Pilipczuk, Rzażewski, and Seymour [1] did not only provide an algorithm for $(\text{tw} \leq d, \psi)$ -MWIS in P_5 -free graphs, but they actually considered a richer class of graphs. In particular, their algorithm works for $C_{>4}$ -free graphs, i.e., graph with no induced cycles of length more than 4 (analogously we define $C_{>t}$ -free graphs for any t). Similarly, the quasipolynomial-time algorithm for $(\text{tw} \leq d, \psi)$ -MWIS works for $C_{>t}$ -free graphs, for any t . Note that $C_{>t}$ -free graphs form a proper superclass of P_t -free graphs. We believe that all polynomial-time results for P_t -free graphs, discussed in this paper, can be actually lifted to $C_{>t}$ -free graphs.

References

- 1 Tara Abrishami, Maria Chudnovsky, Marcin Pilipczuk, Paweł Rzażewski, and Paul D. Seymour. Induced subgraphs of bounded treewidth and the container method. *SIAM J. Comput.*, 53(3):624–647, 2024. doi:10.1137/20M1383732.
- 2 Benjamin Bergougnoux, Édouard Bonnet, Nick Brettell, and O-joung Kwon. Close relatives of feedback vertex set without single-exponential algorithms parameterized by treewidth. In Yixin Cao and Marcin Pilipczuk, editors, *15th International Symposium on Parameterized and Exact Computation, IPEC 2020, December 14–18, 2020, Hong Kong, China (Virtual Conference)*, volume 180 of *LIPICs*, pages 3:1–3:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.IPEC.2020.3.
- 3 Benjamin Bergougnoux, Jan Dreier, and Lars Jaffke. A logic-based algorithmic meta-theorem for mim-width. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22–25, 2023*, pages 3282–3304. SIAM, 2023. doi:10.1137/1.9781611977554.CH125.
- 4 Marthe Bonamy, Nicolas Bousquet, Michał Pilipczuk, Paweł Rzażewski, Stéphan Thomassé, and Bartosz Walczak. Degeneracy of P_t -free and $C_{>t}$ -free graphs with no large complete bipartite subgraphs. *J. Comb. Theory B*, 152:353–378, 2022. doi:10.1016/J.JCTB.2021.10.005.
- 5 Flavia Bonomo, Maria Chudnovsky, Peter Mceli, Oliver Schaudt, Maya Stein, and Mingxian Zhong. Three-coloring and list three-coloring of graphs without induced paths on seven vertices. *Comb.*, 38(4):779–801, 2018. doi:10.1007/S00493-017-3553-8.

- 6 Flavia Bonomo-Braberman, Maria Chudnovsky, Jan Goedgebeur, Peter Maceli, Oliver Schaudt, Maya Stein, and Mingxian Zhong. Better 3-coloring algorithms: Excluding a triangle and a seven vertex path. *Theor. Comput. Sci.*, 850:98–115, 2021. doi:10.1016/J.TCS.2020.10.032.
- 7 Vincent Bouchitté and Ioan Todinca. Listing all potential maximal cliques of a graph. *Theor. Comput. Sci.*, 276(1-2):17–32, 2002. doi:10.1016/S0304-3975(01)00007-X.
- 8 Vincent Bouchitté and Ioan Todinca. Treewidth and minimum fill-in: Grouping the minimal separators. *SIAM J. Comput.*, 31(1):212–232, January 2002. doi:10.1137/S0097539799359683.
- 9 Andreas Brandstädt and Raffaele Mosca. Maximum Weight Independent Sets for $(P_7, \text{triangle})$ -free graphs in polynomial time. *Discret. Appl. Math.*, 236:57–65, 2018. doi:10.1016/J.DAM.2017.10.003.
- 10 Nick Brettell, Jake Horsfield, Andrea Munaro, Giacomo Paesani, and Daniël Paulusma. Bounding the mim-width of hereditary graph classes. *J. Graph Theory*, 99(1):117–151, 2022. doi:10.1002/JGT.22730.
- 11 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.
- 12 Maria Chudnovsky, Jason King, Michał Pilipczuk, Paweł Rzażewski, and Sophie Spirkl. Finding large H -colorable subgraphs in hereditary graph classes. *SIAM J. Discret. Math.*, 35(4):2357–2386, 2021. doi:10.1137/20M1367660.
- 13 Maria Chudnovsky, Rose McCarty, Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzażewski. Sparse induced subgraphs in P_6 -free graphs. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 5291–5299. SIAM, 2024. doi:10.1137/1.9781611977912.190.
- 14 Derek G. Corneil, Yehoshua Perl, and Lorna K. Stewart. A linear recognition algorithm for cographs. *SIAM J. Comput.*, 14(4):926–934, 1985. doi:10.1137/0214065.
- 15 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique width. In Juraj Hromkovic and Ondrej Šýkora, editors, *Graph-Theoretic Concepts in Computer Science, 24th International Workshop, WG '98, Smolenice Castle, Slovak Republic, June 18-20, 1998, Proceedings*, volume 1517 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 1998. doi:10.1007/10692760_1.
- 16 Clément Dallard, Martin Milanič, and Kenny Storgel. Treewidth versus clique number. II. Tree-independence number. *J. Comb. Theory B*, 164:404–442, 2024. doi:10.1016/J.JCTB.2023.10.006.
- 17 Guo Li Ding, Paul Seymour, and Peter Winkler. Bounding the vertex cover number of a hypergraph. *Combinatorica*, 14(1):23–34, March 1994. doi:10.1007/BF01305948.
- 18 Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965. doi:10.4153/CJM-1965-045-4.
- 19 Fedor V. Fomin, Ioan Todinca, and Yngve Villanger. Large induced subgraphs via triangulations and CMSO. *SIAM J. Comput.*, 44(1):54–87, 2015. doi:10.1137/140964801.
- 20 Peter Gartland and Daniel Lokshtanov. Independent set on P_k -free graphs in quasi-polynomial time. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 613–624. IEEE, 2020. doi:10.1109/FOCS46700.2020.00063.
- 21 Peter Gartland, Daniel Lokshtanov, Tomáš Masařík, Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzażewski. Maximum weight independent set in graphs with no long claws in quasi-polynomial time. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 683–691. ACM, 2024. doi:10.1145/3618260.3649791.
- 22 Peter Gartland, Daniel Lokshtanov, Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzażewski. Finding large induced sparse subgraphs in $C_{>t}$ -free graphs in quasipolynomial time. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 330–341. ACM, 2021. doi:10.1145/3406325.3451034.

- 23 Andrzej Grzesik, Tereza Klimošová, Marcin Pilipczuk, and Michał Pilipczuk. Polynomial-time algorithm for maximum weight independent set on P_6 -free graphs. *ACM Trans. Algorithms*, 18(1):4:1–4:57, 2022. doi:10.1145/3414473.
- 24 Martin Grötschel, Laszlo Lovász, and Alexander Schrijver. Polynomial algorithms for perfect graphs. In C. Berge and V. Chvátal, editors, *Topics on Perfect Graphs*, volume 88 of *North-Holland Mathematics Studies*, pages 325–356. North-Holland, 1984. doi:10.1016/S0304-0208(08)72943-8.
- 25 Frédéric Havet, Ross J. Kang, and Jean-Sébastien Sereni. Improper coloring of unit disk graphs. *Networks*, 54(3):150–164, 2009. doi:10.1002/net.20318.
- 26 Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. A near-optimal planarization algorithm. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1802–1811. SIAM, 2014. doi:10.1137/1.9781611973402.130.
- 27 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*, pages 85–103, Boston, MA, 1972. Springer US. doi:10.1007/978-1-4684-2001-2_9.
- 28 Ton Kloks, Ching-Hao Liu, and Sheung-Hung Poon. Feedback vertex set on chordal bipartite graphs. *CoRR*, abs/1104.3915, 2012. doi:10.48550/arXiv.1104.3915.
- 29 Paloma T. Lima, Martin Milanič, Peter Mursic, Karolina Okrasa, Paweł Rzażewski, and Kenny Štorgel. Tree decompositions meet induced matchings: beyond Max Weight Independent Set. *CoRR*, abs/2402.15834, 2024. doi:10.48550/arXiv.2402.15834.
- 30 Daniel Lokshtanov, Martin Vatshelle, and Yngve Villanger. Independent set in P_5 -free graphs in polynomial time. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 570–581. SIAM, 2014. doi:10.1137/1.9781611973402.43.
- 31 Vadim V. Lozin and Dieter Rautenbach. Some results on graphs without long induced paths. *Inf. Process. Lett.*, 88(4):167–171, 2003. doi:10.1016/J.IPL.2003.07.004.
- 32 Vadim V. Lozin and Viktor Zamaraev. The structure and the number of P_7 -free bipartite graphs. *Eur. J. Comb.*, 65:143–153, 2017. doi:10.1016/J.EJC.2017.05.008.
- 33 Pranabendu Misra, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Parameterized algorithms for even cycle transversal. In Martin Charles Golumbic, Michał Stern, Avivit Levy, and Gila Morgenstern, editors, *Graph-Theoretic Concepts in Computer Science - 38th International Workshop, WG 2012, Jerusalem, Israel, June 26-28, 2012, Revised Selected Papers*, volume 7551 of *Lecture Notes in Computer Science*, pages 172–183. Springer, 2012. doi:10.1007/978-3-642-34611-8_19.
- 34 Giacomo Paesani, Daniël Paulusma, and Paweł Rzażewski. Feedback Vertex Set and Even Cycle Transversal for H -free graphs: Finding large block graphs. *SIAM J. Discret. Math.*, 36(4):2453–2472, 2022. doi:10.1137/22m1468864.
- 35 Marcin Pilipczuk. A tight lower bound for Vertex Planarization on graphs of bounded treewidth. *Discret. Appl. Math.*, 231:211–216, 2017. doi:10.1016/j.dam.2016.05.019.
- 36 Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzażewski. Quasi-polynomial-time algorithm for independent set in P_t -free graphs via shrinking the space of induced paths. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 204–209. SIAM, 2021. doi:10.1137/1.9781611976496.23.
- 37 Marcin Pilipczuk and Paweł Rzażewski. A polynomial bound on the number of minimal separators and potential maximal cliques in P_6 -free graphs of bounded clique number. *CoRR*, abs/2310.11573, 2023. doi:10.48550/arXiv.2310.11573.