On the Tractability of Optimization Problems on *H*-Graphs

Fedor V. Fomin[1]
Department of Informatics, University of Bergen, Norway
fedor.fomin@ii.uib.no
https://orcid.org/0000-0003-1955-4612

Petr A. Golovach[2]
Department of Informatics, University of Bergen, Norway
petr.golovach@ii.uib.no
https://orcid.org/0000-0002-2619-2990

Jean-Florent Raymond[3]
Technische Universität Berlin, Germany
raymond@tu-berlin.de
https://orcid.org/0000-0003-4646-7602

Abstract

For a graph H, a graph G is an H-graph if it is an intersection graph of connected subgraphs of some subdivision of H. These graphs naturally generalize several important graph classes like interval graphs or circular-arc graph. This notion was introduced in the early 1990s by Bíró, Hujter, and Tuza. Recently, Chaplick et al. initiated the algorithmic study of H-graphs by showing that a number of fundamental optimization problems like Clique, Independent Set, or Dominating Set are solvable in polynomial time on H-graphs. We extend and complement these algorithmic findings in several directions.

First we show that for every fixed H, the class of H-graphs is of logarithmically-bounded boolean-width. We also prove that H-graphs are graphs with polynomially many minimal separators. Pipelined with the plethora of known algorithms on graphs of bounded boolean-width and graphs with polynomially many minimal separators, this describes a large class of optimization problems that are solvable in polynomial time on H-graphs.

The most fundamental optimization problems among those solvable in polynomial time on H-graphs are Clique, Independent Set, and Dominating Set. We provide a more refined complexity analysis of these problems from the perspective of parameterized complexity. We show that Independent Set and Dominating Set are W[1]-hard being parameterized by the size of H plus the size of the solution. On the other hand, we prove that when H is a tree, Dominating Set is fixed-parameter tractable (FPT) parameterized by the size of H. Besides, we show that Clique admits a polynomial kernel parameterized by H and the solution size.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms, Theory of computation → Graph algorithms analysis

Keywords and phrases H-topological intersection graphs, parameterized complexity, minimal separators, boolean-width, mim-width

Digital Object Identifier 10.4230/LIPIcs.ESA.2018.30

---

[1] Supported by the Research Council of Norway via the project "MULTIVAL".
[2] Supported by the Research Council of Norway via the project "CLASSIS".
[3] Supported by the European Research Council (ERC) via the ERC consolidator grant DISTRUCT-648527. This research was partly done while the author was supported by the Polish National Science Centre grant PRELUDIUM DEC-2013/11/N/ST6/02706.

26th Annual European Symposium on Algorithms (ESA 2018).
Editors: Yossi Azar, Hannah Bast, and Grzegorz Herman; Article No. 30; pp. 30:1–30:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Related Version A full version of the paper is available at [13], https://arxiv.org/abs/1709.09737.

Acknowledgements We thank Saket Saurabh for bringing H-graphs to our attention.

1 Introduction

The notion of H-graph was introduced in the work of Bíró, Hujter, and Tuza [4] on precoloring extensions of graphs. H-graphs nicely generalize several popular and widely studied classes of graphs. For example, the classical definition of an interval graph is as a graph which is an intersection graph[4] of intervals of a line. Equivalently, a graph is interval if it is an intersection graph of some subpaths of a path. Or, equivalently, if it is an intersection graph of some subgraphs of some subdivision (that is a graph obtained by placing vertices of degree 2 on the edges) of P_2, the graph with two adjacent vertices. More generally, for a fixed graph H, an H-graph is an intersection graph of some connected subgraphs of some subdivision of H. Thus for example, an interval graph is a P_2-graph, a circular-arc graph is a C_2-graph, where C_2 is a double-edge with two endpoints, a split graph is a $K_{1,d}$-graph for some $d \geq 0$, where $K_{1,d}$ is a star with d leaves, a chordal graph is a T-graph for some tree T, etc..

The main motivation behind the study of H-graphs is the following. It is well-known that on interval, chordal, circular-arc, and other graphs with "simple" intersection models many NP-hard optimization problems are solvable in polynomial time, see e.g. the book of Golumbic [17] for an overview. It is a natural question whether at least some of these algorithmic results can be extended to more general classes of intersection graphs. Chaplick et al. [8, 9] initiated the systematic study of algorithmic properties of H-graphs. They showed that a number of fundamental optimization problems like Independent Set and Dominating Set are solvable in polynomial time on H-graphs for any fixed H. Most of their algorithms run on H-graphs in time $n^{f(H)}$, where n is the number of vertices in the input graph and f is some function. In other words, being parameterized by H most of the problems are known to be in the class XP. Our work is driven by the following question.

- Are there generic explanations why many problems admit polynomial time algorithms on H-graphs?

We address the first question by proving the following combinatorial results. We show first that every n-vertex H-graph has boolean-width at most $2|E(H)| \cdot \log n$, and that a decomposition of this width can be found in polynomial time. This combinatorial result extends the results of Belmonte and Vatshelle [2, 1] on the boolean-width of interval and circular-arc graphs to H-graphs. Together with the algorithms for a vast class of problems called *LC-VSP* problems [6, 2], and for problems related to induced paths [21], this implies immediately that all these problems are solvable in polynomial time on H-graphs. The illustrative problems where this approach is successful are Weighted Independent Set, Weighted Dominating Set, Total Dominating Set, Induced Matching, Longest Induced Path and Disjoint Induced Paths, and many others.

Then we prove that every n-vertex H-graph has at most $(2n+1)^{|E(H)|} + |E(H)| \cdot (2n)^2$ minimal separators.[5] Pipelining the bound on the number of minimal separators in H-graphs with meta-algorithmic results of Fomin, Todinca and Villanger [14], we obtained another wide class of problems solvable in polynomial time on H-graphs. Examples of such problems

[4] The intersection graph of a family $\mathcal{S}$ of sets has vertex set $\mathcal{S}$ and edge set $\{SS',\ S \cap S' \neq \emptyset\}$.

[5] It was reported to us by Steven Chaplick and Peter Zeman that they also obtained this result independently and that it will be included in the journal version of their paper.

are Treewidth, Feedback Vertex Set, Maximum Induced Subgraph excluding a planar minor, and various packing problems.

All these generic algorithmic results provide XP algorithms when parameterized by the size of H. This brings us to the second question defining the direction of our research.

- What is the parameterized complexity of the fundamental optimization problems being parameterized by the size of H?

The first steps in this direction were done by Chaplick et al. in [8] who showed that Dominating Set is fixed-parameter tractable (FPT) on $K_{1,d}$-graphs parameterized by d. In this paper we show that Dominating Set is W[1]-hard parameterized by the size of H plus the solution size. Thus the existence of an FPT algorithm for a general graph H is very unlikely. (We refer to books [11, 10] for definitions from parameterized complexity and algorithms.) We also prove a similar lower bound for Independent Set parameterized by the size of H plus the solution size. Combined with our combinatorial results, these lower-bounds show that Independent Set and Dominating Set are also W[1]-hard when parameterized by mim-width (a graph parameter to be defined in the corresponding section) of the input and the solution size. The technique we develop to establish lower bounds on H-graphs found applications beyond the topic of this paper [20, 21].

On the positive side, we show that when H is a tree, then Dominating Set is FPT parameterized by the size of H. This significantly extends the result from [8] for stars to arbitrary trees. We actually prove a slightly more general result, namely that Dominating Set is FPT on chordal graphs G parameterized by the leafage of the graph, i.e. the minimum number of leaves in the clique tree of G.

Finally we show that Clique admits a polynomial kernel when parameterized by the size of H plus the solution size. This strengthens the result of Chaplick et al. [8] who showed that Clique is FPT for such a parameterization.

Organization of the paper. We give hereafter the necessary definitions. In Section 2, we upper-bound the boolean-width of H-graphs and provide algorithmic applications. We address minimal separators of H-graphs in Section 3, again with algorithmic consequences. Last, Section 4 contains our results on the parameterized complexity of some classic optimization problems on H-graphs. Due to space constraints, the proofs of the statements marked with (★) are omitted and some other proofs are just sketched. The full details can be found in the complete version of the paper [13].

Definitions. All graphs in this paper are finite, undirected, loopless, and may have multiple edges. If G is a graph, we denote by $|G|$ and $\|G\|$, respectively, its numbers of vertices and edges (counting multiplicities). If $X, Y \subseteq V(G)$, $\overline{X}$ is the complement of X in $V(G)$ (i.e. $\overline{X} = V(G) \setminus X$), $G[X]$ is the subgraph of G induced by the vertices of X, and $G[X, Y]$ is the bipartite subgraph of G induced by those edges that have one endpoint in X and the other in Y. Unless otherwise stated, logarithms are binary.

In this paper H is always a fixed (multi)graph. We say that a graph G is an H-graph if there is a subdivision H' of H and a collection $\mathcal{M} = \{M_v\}_{v \in V(G)}$ (called an *H-representation* or, simply, *representation*) of subsets of $V(H')$, each inducing a connected subgraph, such that G is isomorphic to the intersection graph of $\mathcal{M}$. To avoid confusion, we refer to the vertices of H and H' as *nodes*. We also say that the nodes of H are *branching* nodes of H' and the other nodes are *subdivision* nodes. If v is a vertex of G, then M_v is the *model* of v in the representation $\mathcal{M}$. For every set $A \subseteq V(G)$, we define $M_A = \bigcup_{v \in A} M_v$. For every node u of H', we denote by V_u the set of vertices of G whose model contains u, i.e. $V_u = \{v \in V(G),\ u \in M_v\}$.

2 H-graphs have logarithmic boolean-width

Boolean-width is a graph invariant that has been introduced in [6] and which is related to the number of different neighborhoods along a cut. Belmonte and Vatshelle showed in [2] that n-vertex interval graphs and circular-arc graphs have boolean-width $\mathcal{O}(\log n)$. In this section, we generalize their result by proving that, for any fixed graph H, n-vertex H-graphs have boolean-width $\mathcal{O}(\log n)$. Using the results of [7, 21], we obtain polynomial algorithms for a vast class of optimization problems on H-graphs. Before we proceed with the proofs, we need to introduce some notions specific to this section.

An *induced matching* in a graph G is a set of vertices that induces a disjoint union of edges. If $X \subseteq V(G)$, $\mathrm{mim}(X)$ denotes the maximum number of edges of an induced matching in $G[X, \overline{X}]$.

Let $d \in \mathbb{N}$ and and let $A \subseteq V(G)$. Two subsets $X, Y \subseteq A$ are said to be *d-neighborhood equivalent*, what we denote by $X \equiv^d_A Y$, if $\min(d, |X \cap N(v)|) = \min(d, |Y \cap N(v)|)$ holds for every $v \in \overline{A}$. We write $\mathrm{nec}_d(A)$ for the number of equivalence classes of the relation $\equiv^d_A$.

A *carving decomposition* of a graph G is a pair (T, δ) where T is a full binary rooted tree (that is, every non-leaf vertex has degree 3) and δ is a bijection from the leaves of T to the vertices of G. A carving decomposition (T, δ) is a *caterpillar decomposition* if T can be obtained from a path by adding a vertex of degree one adjacent to every internal vertex. If $w \in V(T)$, we define V_w as the set of vertices of G in bijection with the leaves of the subtree of T rooted at w. We also denote by $\mathrm{mim}(T, \delta)$ (resp. $\mathrm{nec}_d(T, \delta)$, and $\mathrm{boolw}(T, \delta)$) the maximum of $\mathrm{mim}(V_w)$ (resp. $\max\{\mathrm{nec}_d(V_w), \mathrm{nec}_d(\overline{V_w})\}$, and $\log(\mathrm{nec}_1(V_w))$) taken over all $w \in V(T)$.

The *boolean-width* of (T, δ) is the value $\mathrm{boolw}(T, \delta)$ and the boolean-width of G, denoted by $\mathrm{boolw}(G)$, is the minimum boolean-width of a carving decomposition of G.

The following lemma relates maximum induced matchings to neighborhood equivalence.

▶ **Lemma 1** ([2, Lemma 1 and Lemma 2]). *For every n-vertex graph G and $A \subseteq V(G)$,*
1. $\mathrm{mim}(A) \leq k$ *iff for every $S \subseteq A$ there is a $R \subseteq S$ s.t. $R \equiv^1_A S$ and $|R| \leq k$;*
2. $\mathrm{nec}_d(A) \leq n^{d \cdot \mathrm{mim}(A)}$.

Our results on the boolean-width of H-graphs follow from the next result.

▶ **Theorem 2.** *For every n-vertex H-graph G with $n \geq 2$ whose intersection model is given, we can compute in polynomial time a caterpillar decomposition (T, δ) with $\mathrm{mim}(T, \delta) \leq 2\|H\|$.*

Proof. Let F be the subdivision of H in which G can be realized and let $\{M_v\}_{v \in V(G)}$ be the intersection model of G. Let us arbitrarily fix a branching node r of F. Let $v_1, \ldots, v_n$ be an ordering of $V(G)$ by non-decreasing distance of M_{v_i}'s to r.

▶ **Claim 3.** *For every prefix A of $v_1, \ldots, v_n$ and every $S \subseteq A$, there is a set $R \subseteq S$ of size at most $2\|H\|$ such that $R \equiv^1_A S$.*

Proof. Let A be a prefix of $v_1, \ldots, v_n$ and let $S \subseteq A$. Let $M_A = \bigcup_{v \in A} M_v$ and similarly for $M_{\overline{A}}$ and M_S. Let us consider the path P_e corresponding to some edge $e \in E(H)$. Let $x_1, \ldots, x_p$ be the vertices of P_e in the same order.

Let $v \in A$ and notice that since, by definition, $G[M_v]$ is connected, the vertex set $M_v \cap V(P_e)$ induces at most two connected components in P_e. Indeed if $M_v \cap V(P_e)$ induced more than two connected components, then one of them would not contain any endpoint of P_e, and thus this component would not be connected to other vertices of M_v in $G[M_v]$. Let us assume that it induces at least one connected component and let x_i and x_j be the first and last vertices (wrt. the ordering $x_1, \dots, x_p$) of this component. If $\{x_1, \dots, x_{i-1}\}$ is disjoint from $M_{\overline{A}}$, we say that v is a *left-protector* of P_e. If j is maximum among all vertices that protects the left of P_e, then v is a *rightmost left-protector*. (Informally, it extends the most to the right.) Similarly, v is a *right-protector* the right of P_e if $\{x_{j+1}, \dots, x_p\}$ is disjoint from $M_{\overline{A}}$ and is a *leftmost right-protector* if i is minimal.

Let Z_e be a set containing one (arbitrarily chosen) rightmost left-protector and one leftmost right-protector of e if some exist, and let $R = \bigcup_{e \in E(H)} Z_e$. Clearly $|R| \leq 2\|H\|$. Let us now show that $N(S) \cap \overline{A} \subseteq N(R) \cap \overline{A}$. We consider a vertex $u \in N(S) \cap \overline{A}$ and we show that it also belongs to $N(R)$. Let v be a neighbor of u in S. As u and v are adjacent, M_u and M_v have non-empty intersection. Let e be an edge of H such that M_u and M_v meet on P_e, i.e. $M_u \cap M_v \cap V(P_e) \neq \emptyset$. Again, we denote by $x_1, \dots, x_p$ the vertices of P_e.

▶ **Claim 4 (★).** *Let $w \in A$. If $M_w \cap V(P_e) = \{x_i, \dots, x_j\}$ with $1 \leq i \leq j \leq p$, then one of $\{x_k,\ 1 \leq k < i\}$ and $\{x_k,\ j < k \leq p\}$ is disjoint from $M_{\overline{A}}$.*

As M_u intersects M_v on P_e, it intersects the vertex set C of one component induced by M_v on P_e (recall that there are either one or two such components). In the case where there are two components, we assume without loss of generality that this is the “left” one (i.e. that with smallest indices). In the case where there is one component, we assume that v is a left-protector of P_e (according to Claim 4, v is a left-protector or a right-protector of P_e). Observe that in both cases, v is a left-protector of P_e. Let z be the rightmost left-protector of P_e that belongs to R and let $x_k, \dots x_{k'}$ be the vertices of the corresponding component of $P_e[M_z \cap V(P_e)]$ (that is, the component used in the definition of left-protector).

Notice that $C \subseteq \{x_1, \dots, x_{k'}\}$, by maximality of z (informally, because it is “rightmost”). As z is a left-protector, $M_u \cap \{x_1, \dots, x_{k-1}\} = \emptyset$. Since M_u and C intersect, they intersect in $\{x_k, \dots, x_{k'}\}$. Therefore $M_u \cap M_z \neq \emptyset$: z is adjacent to u. As $z \in R$, we are done. ◀

We construct a caterpillar decomposition that follows the ordering $v_1, \dots, v_n$. If $n = 2$, then we define T to be the tree with the two vertices and δ maps them to v_1 and v_2. Assume that $n \geq 3$. We construct a path $x_2 \dots x_{n-1}$ and n vertices $y_1, \dots, y_n$. Then we make y_1, y_2 adjacent to x_2, y_i is made adjacent to x_i if $3 \leq i \leq n-2$, and y_{n_1}, y_n is adjacent to x_{n-1}. We define $\delta(y_i) = v_i$ for $i \in \{1, \dots, n\}$. In both cases the root is chosen arbitrarily. According to Claim 3 and Lemma 1.(1), this caterpillar decomposition satisfies $\operatorname{mim}(T, \delta) \leq 2\|H\|$. ◀

The next result follows from the application to the decomposition provided by Theorem 2 of Lemma 1.(2), with the fact that $\operatorname{mim}(A) = \operatorname{mim}(\overline{A})$ for every $A \subseteq V(G)$.

▶ **Corollary 5.** *For every n-vertex H-graph G with $n \geq 2$ whose intersection model is given, we can compute in polynomial time a caterpillar decomposition (T, δ) with $\operatorname{nec}_d(T, \delta) \leq n^{d \cdot 2\|H\|}$.*

From the definition of boolean-width, we also get:

▶ **Corollary 6.** *Every n-vertex H-graph with $n \geq 2$ has boolean-width at most $2\|H\| \cdot \log n$.*

By choosing H to be a single or double edge, we recover the results of [2] on the boolean-width of interval and circular-arc graphs, respectively, as special cases of Corollary 6. Apart of the degenerate case where H is edgeless (in which case H-graphs are disjoint unions of cliques), every interval graph is an H-graph. Hence the $\Omega(\log n)$ lower bound of [2] shows that Corollary 6 is tight up to a constant factor.

Algorithmic applications. Boolean-width and nec_d have been used in [6, 7] to design parameterized algorithms for the problems Weighted Independent Set, Weighted Dominating Set, and a vast class of problems, called *LC-VSP* problems, that includes fundamental problems as Independent Set, Independent Dominating Set, Total Dominating Set, Induced Matching, and many others (see [7]). The main result of [7] is the following.

▶ **Theorem 7** ([7]). *For every LC-VSP problem* Π*, there are constants* d *and* q *such that* Π *can be solved in time* $\mathcal{O}(n^4 \cdot q \cdot \text{nec}_d(T,\delta)^{3q})$ *if a decomposition* (T,δ) *of the input is given.*

Recently, Jaffke, Kwon, and Telle obtained polynomial-time algorithms on graphs of bounded mim-width for problems that are not LC-VSP.

▶ **Theorem 8** ([21]). *The problems* Longest Induced Path*,* Induced Disjoint Paths*, and* H-Induced Subdivision[6] *can be solved in time* $n^{\mathcal{O}(\text{mim}(T,\delta))}$ *if a decomposition* (T,δ) *of the input is given.*

Combining Theorem 2 with Theorem 8 and Corollary 5 with Theorem 7, we get the following meta-algorithmic consequences.

▶ **Theorem 9.** *Let* H *be a graph and let* Π *be either a LC-VSP problem or one of* Longest Induced Path*,* Induced Disjoint Paths*, and* H-Induced Subdivision*. Then* Π *can be solved in polynomial time on* H*-graphs if an* H*-representation of the input is provided.*

3 H-graphs have few minimal separators

Let G be a graph. A set $X \subseteq V(G)$ is a *minimal separator* of G if $G \setminus X$ has more connected components than G, and X is inclusion-minimal with this property. The study of minimal separators is an active line of research that found many algorithmic applications (see e.g. [22, 3, 5, 14]). In general, the number of minimal separators of a graph may be as large as exponential in its number of vertices. We prove in this section that in an H-graph, this number is upper-bounded by a polynomial (Theorem 10). Combining this finding with the meta-algorithmic results of [14], we deduce that a wide class of optimization problems can be solved in polynomial time on H-graphs (Corollary 16).

▶ **Theorem 10.** *Every* H*-graph* G *has at most* $(2|G|+1)^{\|H\|} + \|H\| \cdot (2|G|)^2$ *minimal separators.*

Proof (sketch). Let G be a H-graph and let F be a subdivision of H where G can be represented as the intersection graph of $\{M_v,\ v \in V(G)\}$. For every subset $V \subseteq V(G)$, the *border edges* of V are the edges of F with one endpoint in M_V and one endpoint in $V(F) \setminus M_V$. Let R be the union of border edges over $\{M_v,\ v \in V(G)\}$. Observe that for every $V \subseteq V(G)$, the set of border edges of V is a subset of R. For every $S \subseteq E(F)$, let V_S be the set of all vertices of G whose model contain some edge of S.

▶ **Claim 11** (★). *For every minimal separator* X *in* G*, there is a* $S \subseteq R$ *such that* $X = V_S$*.*

From Claim 11 we can already deduce that the number of minimal separators of G is at most the number of subsets of R. To get better bounds, we need other observations. The next claim follows from the fact that $F[M_V]$ is connected.

[6] We refer the reader to [21] for an accurate definition of these problems.

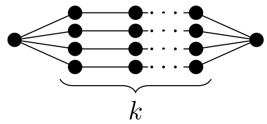

Figure 1 A θ_4-graph.

▶ **Claim 12.** *For every $V \subseteq V(G)$ such that M_V induces a connected subgraph of F, and every $e \in E(H)$, the set M_V has at most two border edges in $E(P_e)$. Hence, $|R| \leq 2|G| \cdot \|H\|$.*

▶ **Claim 13.** *For every minimal separator X of G, if $S \subseteq R$ is the subset of edges of F defined in the proof of Claim 11, then either $|S| = 2$ and $S \subseteq E(P_e)$ for some $e \in E(H)$, or $|S \cap E(P_e)| \leq 1$ for every $e \in E(H)$.*

Proof (sketch). Let A, B be two connected components of $G \setminus X$ such that $N(A) = N(B) = X$. From Claim 12 we get $|S \cap E(P_e)| \leq 2$ for every $e \in E(H)$. Intuitively, if $|S \cap E(P_e)| = 2$ for some $e \in E(H)$ then one of M_A and M_B contains only interior vertices of P_e. From the definition of S we deduce $|S| = 2$. ◀

Therefore, for every minimal separator X of G, there is a set $S \subseteq R$ such that:
1. either $|S \cap E(P_e)| \leq 1$ for every $e \in E(H)$;
2. or $|S| = 2$ and $S \subseteq E(P_e)$ for some $e \in E(H)$;

In order to upper-bound the number of possible minimal separators of G, we can consequently upper-bound the number of sets $S \subseteq R$ that satisfy one of the two conditions above, and (using Claim 12) obtain the bound of $(2|G| + 1)^{\|H\|} + \|H\| \cdot (2|G|)^2$. ◀

For every $r \in \mathbb{N}$, let θ_r be the graph with 2 vertices and r parallel edges. Lemma 14 shows that the exponential contribution of $\|H\|$ in Theorem 10 cannot be avoided. Figure 1 shows an example of a θ_4-graph as in its proof, with at least k^4 minimal separators.

▶ **Lemma 14** (★)**.** *For every $r \in \mathbb{N}$, there is a θ_r-graph G with at least $\left(\frac{|G|-2}{r}\right)^r$ minimal separators.*

Algorithmic applications. Let us consider the following generic problem described in [14].

> Optimal Induced Subgraph for $\mathcal{P}$ and t (OIS($\mathcal{P}, t$) for short)
> **Input:** A graph G;
> **Task:** Find sets $X \subseteq F \subseteq V(G)$ such that X is of maximum size, the induced subgraph $G[F]$ is of treewidth at most t, and $\mathcal{P}(G[F], X)$ is true.

Fomin, Todinca, and Villanger proved that when the property $\mathcal{P}$ can be expressed in Counting Monadic Second Order logic (CMSOL, see [14]), the problem OIS($\mathcal{P}, t$) can be easily solved on classes of graphs that have a polynomial number of minimal separators. This includes natural optimization problems like Treewidth, Feedback Vertex Set, Maximum Induced Subgraph excluding a planar minor, and various packing problems.

▶ **Theorem 15** ([14])**.** *For any fixed t and CSMO property $\mathcal{P}$, OIS($\mathcal{P}, t$) on an n-vertex graph G with s minimal separators, is solvable in time $\mathcal{O}(s^2 \cdot n^{t+4} \cdot f(t, \mathcal{P}))$, for some function f of t and $\mathcal{P}$ only. In particular, the problem is solvable in polynomial time for classes of graphs whose number of minimal separator is upper-bounded by a polynomial function of their order.*

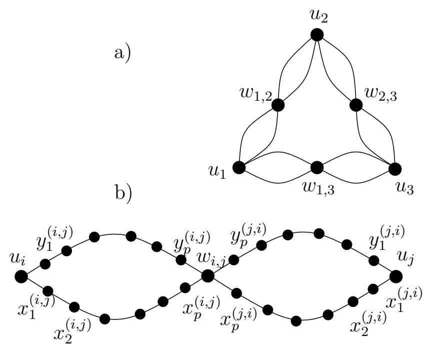

Figure 2 The construction of H for $k = 3$ and the subdivision of its edges.

We deduce the following result about H-graphs.

▶ **Corollary 16.** *Let H be a graph. For any fixed t and CSMO property $\mathcal{P}$,* $\text{OIS}(\mathcal{P}, t)$ *can be solved in polynomial time $\mathcal{O}(n^{\mathcal{O}(|V(H)|)} \cdot n^{t+4} \cdot f(t, \mathcal{P}))$ on H-graphs.*

4 Parameterized complexity of basic problems for H-graphs

In this section we investigate the parameterized complexity of some basic graph problems for H-graphs: Dominating Set, Independent Set and Clique.

4.1 Hardness of of Dominating Set and Independent Set for H-graphs

Recall that Dominating Set and Independent Set, given a graph G and a positive integer k, ask whether G has a dominating set of size at most k and independent set of size at least k respectively. In this section we prove W[1]-hardness of Dominating Set and Independent Set for H-graphs (Theorem 17). Our proofs use a reduction from the Multicolored Clique problem. This problem, given a graph G and a k-partition $V_1, \dots, V_k$ of $V(G)$, asks whether G has a k-clique with exactly one vertex in each V_i for $i \in \{1, \dots, k\}$. The problem is well-known to be W[1]-complete when parameterized by k [12, 23].

▶ **Theorem 17.** Dominating Set *and* Independent Set *are* W[1]*-hard for H-graphs when parameterized by $k + \|H\|$, even if an H-representation of G is given.*

Proof (sketch). Let us show the W[1]-hardness for Independent Set. Let $(G, V_1, \dots, V_k)$ be an instance of Multicolored Clique. We assume that $k \geq 2$ and $|V_i| = p$ for $i \in \{1, \dots, k\}$. Denote by $v_1^i, \dots, v_p^i$ the vertices of V_i for $i \in \{1, \dots, k\}$.

We construct the multigraph H as follows:

(i) Construct k nodes $u_1, \dots, u_k$.

(ii) For every $1 \leq i < j \leq k$, construct a node $w_{i,j}$ and two pairs of parallel edges $u_i w_{i,j}$ and $u_j w_{i,j}$.

(See Figure 2 a).) Note that $|V(H)| = k(k+1)/2$ and $|E(H)| = 2k(k-1)$.

Then we construct the subdivision H' of H obtained by subdividing each edge p times. We denote the subdivision nodes for 4 edges of H constructed for each pair $0 \leq i < j \leq k$ in (ii) by $x_1^{(i,j)}, \dots, x_p^{(i,j)}$, $y_1^{(i,j)}, \dots, y_p^{(i,j)}$, $x_1^{(j,i)}, \dots, x_p^{(j,i)}$ and $y_1^{(j,i)}, \dots, y_p^{(j,i)}$ as it is shown in Figure 2 b).

To simplify notations, we assume that $u_i = x_0^{(i,j)} = y_0^{(i,j)}$, $u_j = x_0^{(j,i)} = y_0^{(j,i)}$ and $w_{i,j} = x_{p+1}^{(i,j)} = y_{p+1}^{(i,j)} = x_{p+1}^{(j,i)} = y_{p+1}^{(j,i)}$. Now we construct the H-graph G' by defining its

H-representation $\mathcal{M} = \{M_v\}_{v\in V(G')}$ where the model of each vertex is a connected subset of $V(H')$. Recall that G is the graph of the original instance of Multicolored Clique.

1. For each $i \in \{1,\ldots,k\}$ and $s \in \{1,\ldots,p\}$, construct a vertex z^i_s with the model

$$M_{z^i_s} = \cup_{j\in\{1,\ldots,k\},j\neq i}(\{x^{(i,j)}_0,\ldots,x^{(i,j)}_{s-1}\} \cup \{y^{(i,j)}_0,\ldots,y^{(i,j)}_{p-s}\}).$$

2. For each edge $v^i_s v^j_t \in E(G)$ for $s,t \in \{1,\ldots,p\}$ and $1 \leq i < j \leq k$, construct a vertex $r^{(i,j)}_{s,t}$ with the model

$$M_{r^{(i,j)}_{s,t}} = (\{x^{(i,j)}_s,\ldots,x^{(i,j)}_{p+1}\}) \cup (\{y^{(i,j)}_{p-s+1},\ldots,y^{(i,j)}_{p+1}\}) \cup (\{x^{(j,i)}_t,\ldots,x^{(j,i)}_{p+1}\}) \cup (\{y^{(j,i)}_{p-t+1},\ldots,y^{(j,i)}_{p+1}\}).$$

Finally, we define $k' = k(k+1)/2$. We claim that $(G, V_1,\ldots,V_k)$ is a yes-instance of Multicolored Clique if and only if G' has an independent set of size k'. The proof is based on the following crucial property of our construction, that can be easily checked.

▶ **Claim 18.** *For every $0 \leq i < j \leq k$, a vertex $z^i_h \in V(G')$ (a vertex $z^j_h \in V(G')$) is not adjacent to a vertex $r^{(i,j)}_{s,t} \in V(G')$ corresponding to the edge $v^i_s v^j_t \in E(G)$ if and only if $h = s$ ($h = t$, respectively).*

Let $\{v^1_{h_1},\ldots,v^k_{h_k}\}$ be a clique of G. Consider the set $I = \{z^1_{h_1},\ldots,z^k_{h_k}\} \cup \{r^{(i,j)}_{h_i,h_j} \mid 0 \leq i < j \leq k\}$ of vertices of G'. It is straightforward to verify using Claim 18 that I is an independent set of size k' in G'. Suppose now that G' has an independent set I of size k'. For each $i \in \{1,\ldots,k\}$, the set $Z_i = \{z^i_h \mid 1 \leq h \leq p\}$ is a clique of G', and for each $1 \leq i < j \leq k$, the set $R_{i,j} = \{r^{(i,j)}_{s,t} \mid 1 \leq s,t \leq p, v^i_s v^j_t \in E(G)\}$ is also a clique of G'. Since all these $k + \binom{k}{2} = k(k+1)/2 = k'$ cliques form a partition of $V(G')$, we have that for each $i \in \{1,\ldots,k\}$, there is a unique $z^i_{h_i} \in Z_i \cap I$, and for every $1 \leq i < j \leq k$, there is a unique $r^{(i,j)}_{s_i,s_j} \in R_{i,j} \cap I$. Since $r^{(i,j)}_{s_i,s_j}$ is not adjacent to $z^i_{h_i}$ and $z^j_{h_j}$, we obtain that $s_i = h_i$ and $s_j = h_j$ by Claim 18. It implies that $v^i_{h_i} v^j_{h_j} \in E(G)$. Since it holds for every $1 \leq i < j \leq k$, $\{v^1_{h_1},\ldots,v^k_{h_k}\}$ is a clique in G.

This completes the W[1]-hardness proof for Independent Set. Our proof can be modified to show the W[1]-hardness of Dominating Set. ◀

We proved in Theorem 2 that for every fixed H, every H-graph has mim-width at most $2\|H\|$. We deduce from the negative results above the following corollary.

▶ **Corollary 19.** *Dominating Set and Independent Set are W[1]-hard when parameterized by the solution size plus the mim-width of the input.*

We note that the construction in the proof of Theorem 17 has been adapted in [19] to show that the Feedback Vertex Set problem is W[1]-hard on H-graphs when parameterized by the solution size plus the number of edges of H.

4.2 Dominating Set for T-graphs

In this section we show that Dominating Set is FPT for chordal graphs when the problem is parameterized by the *leafage* of the input graph, that is, by the minimum number of leaves in a clique tree for the input graph. This result is somehow tight since Dominating Set is well-known to be W[2]-hard for split graphs when parameterized by the solution size [24]. Recall also that Independent Set is polynomial-time solvable for chordal graphs [15, 17] and, therefore, for H-graphs if H is a tree.

Let G be a graph. As it is standard, we say that $u \in V(G)$ (resp. $D \subseteq V(G)$) *dominates* $v \in V(G)$ if $v \in N_G[u]$ (resp. $v \in N_G[D]$) and u (resp. D) dominates a set $W \subseteq V(G)$ if every vertex of W is dominated by u (resp. some vertex of D). Let $\mathcal{K}$ be the set of (inclusion-wise) maximal cliques of G and let $\mathcal{K}_v \subseteq \mathcal{K}$ be the set of maximal cliques containing $v \in V(G)$. A tree T whose node set is $\mathcal{K}$ such that $\mathcal{K}_v$ induce a subtree of T for every $v \in V(G)$ is called a *clique tree* of G. It is well-known [16] that G is a chordal graph if and only if G has a clique tree T. Moreover, if T is a clique tree of G, then G is an intersection graph of subtrees of T, that is, G is a T-graph. Note that a clique tree of a chordal graph is not necessarily unique. For a connected chordal graph G, the *leafage* $\ell(G)$ of G is the minimum number of leaves in its clique tree. It was shown by Habib and Stacho in [18] that given a connected chordal graph G, we can construct in polynomial time its clique tree T with $\ell(G)$ leaves and a T-representation of G.

Let T be a tree and let G be a connected T-graph with its T-representation $\mathcal{M} = \{M_v\}_{v \in V(G)}$ with respect to a subdivision T' of T. For nonempty $U \subseteq V(T)$, we say that $v \in V(G)$ is a *U-vertex* if $M_v \cap V(T) = U$. If $U = \{u\}$, we write u-vertex instead of $\{u\}$-vertex. We denote the set of U-vertices by $V_G(U)$ and $V_G(u)$ if $U = \{u\}$. We also denote by $V_G(T)$ the set of all U-vertices of G for all nonempty $U \subseteq V(T)$. For $e \in E(T)$, $v \in V(G)$ is an e-vertex if M_v contains only subdivision nodes of T' from the path in T' corresponding to e in T. The set of e-vertices is denoted by $V_G(e)$.

We use the following lemma to upper bound the number of vertices in a minimum dominating set whose models contain given nodes of T.

▶ **Lemma 20 (★).** *Let D be a minimum dominating set of G. Let $X \subseteq V(T)$ be an inclusion maximal set of nodes of T such that i) for every $x \in X$, there is $u \in D$ with $x \in M_u$ and ii) for every $xy \in E(T)$ with $x, y \in X$, there is $u \in D$ with $x, y \in M_u$. Then the set $U = \{u \in D \mid X \cap M_u \neq \emptyset\}$ contains at most $|N_T[X]|$ vertices.*

In particular, since $|N_T(X)|$ is at most the number of leaves ℓ, we have that $|U| \leq |X| + \ell$. We use Lemma 20 to obtain an upper bound for the number of vertices in a minimum dominating set whose models contain nodes of T. The next lemma is crucial for our algorithm as it allows to restrict the choice of vertices in a dominating set whose models contain branching vertices. Observe that the models of other vertices form a union of disjoint interval graphs.

▶ **Lemma 21 (★).** *Let D be a minimum dominating set of G. Then $|D \cap V_G(T)| \leq 3|V(T)| - 2$.*

We consider the following auxiliary problem for T-graphs.

Dominating Set Extension
Input: A tree T and a graph G with a given T-representation, positive integers k and d, a *labeling* function $c\colon \bigcup_{x \in V(T)} V_G(x) \to \mathbb{N}$, and a collection of sets $\{C_x\}_{x \in V(T)}$ of size at most d where each $C_x \subseteq c(V_G(x))$ (some sets could be empty) such that for every dominating set D of G of minimum size with the properties that
1. D has at most d x-vertices for $x \in V(T)$, and
2. for each $x \in V(T)$, $C_x \subseteq c(D \cap V_G(x))$,

it holds that the number of nodes $x \in V(T)$ such that D contains an x-vertex is maximum and for each $x \in V(T)$, $C_x = c(D \cap V_G(x))$.
Task: Decide whether there is a dominating set D' of G of size at most k containing at most d x-vertices for $x \in V(T)$ such that for each $x \in V(T)$, $C_x = c(D' \cap V_G(x))$.

Note that Dominating Set Extension is a *promise* problem: we are promised that there is D with the described properties but D itself is not given. Moreover, the promise could be false but we are not asked to verify it. The labeling c in the statement of the problem and the promise define, in fact, the choice of the vertices in a dominating set whose models contain branching vertices.

We use dynamic programming to solve this problem, but we are solving it only for graphs with special representations. Let $\mathcal{M} = \{M_v\}_{v \in V(G)}$ be a T-representation of G with respect to a subdivision T'. We say that $\mathcal{M}$ is *nice* if $|M_v \cap V(T)| \leq 1$ for $v \in V(G)$, i.e., each set M_v contains at most one branching node of T'. This considerably simplifies handling of the vertices whose models contain branching nodes that are selected to be included in a dominating set by our dynamic programming algorithm. As it is standard for dynamic programming, we pick a root r in T that defines the parent-child relation on $V(T)$ and $V(T')$.

▶ **Lemma 22** (★). *Given a nice r-rooted T-representation of the input graph where T is a tree with ℓ leaves,* Dominating Set Extension *can be solved in time $2^{\mathcal{O}((d+\ell)\log d)} n^{\mathcal{O}(1)}$. Moreover, it can be done by an algorithm that either returns a correct yes-answer or (possible incorrect) no-answer even if the promise is false.*

To be able to make a given T-representation nice, we define contractions of edges of T that transforms G as well. For an edge $e \in E(T)$, we say that G' is obtained *by contracting e in T* if G' is the (T/e)-graph with the model obtained as follows:

1. contract xy in T and, respectively, the (x, y)-path P in T', and denote the node obtained from x and y by z,
2. delete all e-vertices of G,
3. for each remaining vertex $u \in V(G)$, delete from M_u the subdivision nodes of P and replace x and y by z if at least one of these nodes is in M_u.

Note that $V(G') \subseteq V(G)$ and $G[V(G')]$ is a subgraph of G' but not necessarily induced since two vertices of G' that are not adjacent in G could be adjacent in G'.

Now we are ready to explain how we solve Dominating Set for chordal graphs of bounded leafage.

▶ **Theorem 23.** Dominating Set *can be solved in time $2^{\mathcal{O}(\ell^2)} \cdot n^{\mathcal{O}(1)}$ for connected chordal graphs with leafage at most ℓ.*

Proof (sketch). Let (G, k) be an instance of Dominating Set where G is a connected chordal graph.

We use the algorithm of Habib and Stacho [18] to compute its leafage $\ell(G)$. If $\ell(G) > \ell$, we stop and return a no-answer. Otherwise, we consider the clique tree T' of G constructed by the algorithm. We construct the tree T from T' by *dissolving* nodes of degree two, that is, for a node x of degree two with the neighbors y and z, we delete x and make y and z adjacent. Observe that since T is a tree with at most ℓ leaves that has no node of degree two, $|V(T)| \leq 2\ell - 2$. We have that G is a T-graph. Note also that the algorithm of Habib and Stacho [18] gives us a T-representation $\mathcal{M} = \{M_v\}_{v \in V(G)}$ where $M_v \in V(T')$ for $v \in V(G)$.

We consider $2^{|V(T)|} - 1 \leq 2^{2\ell-2} - 1$ nonempty subsets of $V(T)$ and construct a *coloring* $c\colon V_G(T) \to \{1, \ldots, 2^{|V(T)|}\}$ such that for $u, v \in V_G(T)$, $c(u) = c(v)$ if and only if u and v are U-vertices for the same $U \subseteq V(T)$.

By Lemma 21, a minimum dominating set of G contains at most $3|V(T)| - 2 \leq 6\ell - 8$ vertices of $V_G(T)$. Clearly, these vertices can have at most $6\ell - 8$ distinct colors. We consider all sets $C \subseteq \{1, \ldots, 2^{|V(T)|}\}$ of distinct colors of size at most $6\ell - 8$ and for each C, we aim to find a minimum dominating set of G whose vertices in $V_G(T)$ are colored by the maximum number of distinct colors and are colored exactly by the colors of C. Since we consider all possible choices of C, it holds for some C. Toward this aim, we apply the following rule.

Rule 1. If there is an xy-vertex w of G for $xy \in E(T)$ such that i) $x, y \notin M_u$ for $u \in V_G(T)$ with $c(u) \in C$ and ii) there is $v \in V_G(T)$ such that $x, y \in M_v$, then discard the choice of C.

Now we are looking for a dominating set D of minimum size such that $c(D \cap V_G(T)) = C$. We use the following rule.

Rule 2. If there is a U-vertex u of G for nonempty $U \subseteq V(T)$ such that i) $c(u) \notin C$ and ii) there is $c \in C$ such that for every $v \in V_G(T)$ with $c(v) = c$, v dominates u, then delete u.

Our next aim is to construct a nice representation. Let

$$A = \{xy \in E(T) \mid x, y \in M_u \text{ for some } u \in V_G(T) \text{ such that } c(u) \in C\},$$
$$A' = \{xy \in E(T) \mid x, y \in M_u \text{ for some } u \in V_G(T) \text{ such that } c(u) \notin C \text{ and } x, y \notin M_v \text{ for } v \in V_G(T) \text{ such that } c(v) \in C\}.$$

Because of Rule 1, there is no e-vertex for $e \in A'$. We contract the edges $e \in A \cup A'$. Let $\hat{T}$ (resp. $\hat{T}'$) be the tree obtained from T (resp. T') by contracting the paths that correspond to the contracted edge. We also construct the graph $\hat{G}$ that is obtained from G by contracting these edges of T and we also construct its $\hat{T}$-representation $\hat{\mathcal{M}} = \{\hat{M}_v\}_{v \in V(\hat{G})}$ where $\hat{M}_v \in V(\hat{T}')$ for $v \in V(\hat{G})$. We set $\hat{c} = c|_{V(\hat{G})}$ and let $C_x = \{c, \exists u \in V_{\hat{G}}(x) \text{ s.t. } \hat{c}(u) = c\}$ for $x \in V(\hat{T})$. Observe that $\hat{\mathcal{M}}$ is a nice $\hat{T}$-representation of $\hat{G}$. Indeed, for every $xy \in E(T)$ such that $x, y \in M_u$ for $u \in V_G(T)$ we have that $xy \in A$ if $c(u) \in C$ and $xy \in A'$ if $c(u) \notin C$ because of Rule 2, and all such edges xy are contracted.

We show that the contraction of the edges of $A \cup A'$ is safe in the sense that D is a minimum dominating set with $C = c(D \cap V_G(T))$ if and only if D is a minimum dominating set of G'such that $C = \hat{c}(D \cap V_{\hat{G}}(\hat{T}))$. Note that the condition $C = \hat{c}(D \cap V_{\hat{G}}(\hat{T}))$ is equivalent to the condition that for every $x \in V(\hat{T})$, $C_x = \hat{c}(D \cap V_{\hat{G}}(x))$, because the $\hat{T}$-representation of $\hat{G}$ is nice and $C_x \cap C_y = \emptyset$ for distinct $x, y \in V(\hat{T})$. We set $d = |V(T)| + \ell \leq 3\ell - 2$ and apply the next rule.

Rule 3. If there is $x \in V(\hat{T})$ with $|C_x| > d$, then discard the current choice of C.

By Lemma 20, we have that if a set of nodes X of T is contracted into a single vertex x of $\hat{T}$, then D has at most $|X| + \ell$ vertices whose models contain a vertex of X and, therefore, the number of vertices colored by the colors of C_x in D is at most d.

We select arbitrarily a node r to be the root of $\hat{T}$ and $\hat{T}'$ respectively. Then we apply Lemma 22 for the instance $(\hat{T}, k, d, \hat{c}, \{C_x\}_{x \in V(\hat{T})})$ of Dominating Set Extension.

Recall that Dominating Set Extension is a promise problem. If the algorithm from Lemma 22 returns a yes-answer, it means that there is a dominating set D of $\hat{G}$ of size at most k such that for each $x \in V(\hat{T})$, $C_x = c(D \cap V_{\hat{G}}(x))$. It means that the input graph G has a dominating set of size at most k. Still, if the promise is false, the algorithm can return an incorrect no-answer. Recall that the promise of Dominating Set Extension is the following: for every dominating set D of $\hat{G}$ of minimum size with the properties that

1. D has at most d x-vertices for $x \in V(\hat{T})$, and
2. for each $x \in V(\hat{T})$, $C_x \subseteq c(D \cap V_G(\hat{x}))$,

it holds that the number of nodes $x \in V(\hat{T})$ such that D contains an x-vertex is maximum and for each $x \in V(\hat{T})$, $C_x = c(D \cap V_G(\hat{x}))$. We prove that if C is chosen in such a way that G has a minimum dominating set D that has the maximum number of vertices of $V_G(T)$ and whose vertices in $V_G(T)$ are colored exactly by the colors of C, then this promise holds for the corresponding instance of Dominating Set Extension constructed for this choice of C. Therefore, if (G, k) is a yes-instance of Dominating Set, then for some choice of C, we obtain a yes-answer. ◀

The theorem immediately gives the following corollary for T-graphs.

▶ **Corollary 24.** Dominating Set *can be solved in time* $2^{\mathcal{O}(|T|^2)} \cdot n^{\mathcal{O}(1)}$ *for* T*-graphs if* T *is a tree.*

4.3 A polynomial kernel for Clique

It was observed in [8] that the Clique problem is FPT for H-graphs when parameterized by the solution size k and $\|H\|$ (even when no H-representation of G is given). We show that Clique admits a polynomial kernel when a representation is given.

▶ **Theorem 25 (★).** *The* Clique *problem for* H*-graphs admits a kernel with at most* $(k-1)|V(H)|$ *vertices if an* H*-representation of the input graph is given.*

References

1 Rémy Belmonte and Martin Vatshelle. On graph classes with logarithmic boolean-width. *arXiv preprint*, 2010. arXiv:1009.0216.

2 Rémy Belmonte and Martin Vatshelle. Graph classes with structured neighborhoods and algorithmic applications. *Theoretical Computer Science*, 511:54–65, 2013. doi:10.1016/j.tcs.2013.01.011.

3 Anne Berry, Jean-Paul Bordat, and Olivier Cogis. *Generating All the Minimal Separators of a Graph*, pages 167–172. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999. doi:10.1007/3-540-46784-X_17.

4 M. Biró, M. Hujter, and Zs. Tuza. Precoloring extension. I. Interval graphs. *Discrete Mathematics*, 100(1):267–279, 1992. doi:10.1016/0012-365X(92)90646-W.

5 Vincent Bouchitté and Ioan Todinca. Treewidth and minimum fill-in: Grouping the minimal separators. *SIAM Journal on Computing*, 31(1):212–232, 2001. doi:10.1137/S0097539799359683.

6 Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Boolean-width of graphs. *Theoretical Computer Science*, 412(39):5187–5204, 2011. doi:10.1016/j.tcs.2011.05.022.

7 Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems. *Theoretical Computer Science*, 511:66–76, 2013. Exact and Parameterized Computation. doi:10.1016/j.tcs.2013.01.009.

8 Steven Chaplick, Martin Töpfer, Jan Voborník, and Peter Zeman. On H-topological intersection graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 167–179. Springer, 2017. arXiv:1608.02389.

9 Steven Chaplick and Peter Zeman. Combinatorial problems on H-graphs. In *The European Conference on Combinatorics, Graph Theory and Applications (EUROCOMB'17), Electronic Notes in Discrete Mathematics*, volume 61, pages 223–229, 2017. arXiv:1706.00575. doi:10.1016/j.endm.2017.06.042.

10 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Dániel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

11 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

12 Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009. doi:10.1016/j.tcs.2008.09.065.

13 Fedor V. Fomin, Petr A. Golovach, and Jean-Florent Raymond. On the tractability of optimization problems on h-graphs. *CoRR*, abs/1709.09737, 2017. arXiv:1709.09737.

14 Fedor V. Fomin, Ioan Todinca, and Yngve Villanger. Large induced subgraphs via triangulations and CMSO. *SIAM J. Comput.*, 44(1):54–87, 2015. arXiv:1309.1559. doi:10.1137/140964801.

15 Fănică Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM J. Comput.*, 1(2):180–187, 1972. doi:10.1137/0201013.

16 Fănică Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *J. Combinatorial Theory Ser. B*, 16:47–56, 1974.

17 Martin Charles Golumbic. *Algorithmic graph theory and perfect graphs*, volume 57 of *Annals of Discrete Mathematics*. Elsevier Science B.V., Amsterdam, second edition, 2004. With a foreword by Claude Berge.

18 Michel Habib and Juraj Stacho. Polynomial-time algorithm for the leafage of chordal graphs. In *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, volume 5757 of *Lecture Notes in Computer Science*, pages 290–300. Springer, 2009. doi:10.1007/978-3-642-04128-0_27.

19 L. Jaffke, O. Kwon, and J. A. Telle. A note on the complexity of Feedback Vertex Set parameterized by mim-width. *arXiv preprint*, 2017. arXiv:1711.05157. arXiv:1711.05157.

20 Lars Jaffke, O-joung Kwon, and Jan Arne Telle. A note on the complexity of feedback vertex set parameterized by mim-width. *CoRR*, abs/1711.05157, 2017. arXiv:1711.05157.

21 Lars Jaffke, O-joung Kwon, and Jan Arne Telle. Polynomial-time algorithms for the longest induced path and induced disjoint paths problems on graphs of bounded mim-width. *arXiv preprint*, 2017. arXiv:1708.04536.

22 Ton Kloks, H Bodlaender, Haiko Müller, and Dieter Kratsch. Computing treewidth and minimum fill-in: All you need are the minimal separators. *Algorithms—ESA'93*, pages 260–271, 1993.

23 Krzysztof Pietrzak. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *J. Comput. Syst. Sci.*, 67(4):757–771, 2003. doi:10.1016/S0022-0000(03)00078-3.

24 Venkatesh Raman and Saket Saurabh. Short cycles make W[1]-hard problems hard: FPT algorithms for W[1]-hard problems in graphs with no short cycles. *Algorithmica*, 52(2):203–225, 2008. doi:10.1007/s00453-007-9148-9.