Tight Bounds for Chordal/Interval Vertex Deletion Parameterized by Treewidth

Michał Włodarczyk
Ben-Gurion University, Beer Sheva, Israel

Abstract

In CHORDAL/INTERVAL VERTEX DELETION we ask how many vertices one needs to remove from a graph to make it chordal (respectively: interval). We study these problems under the parameterization by treewidth $\mathbf{tw}$ of the input graph G. On the one hand, we present an algorithm for CHORDAL VERTEX DELETION with running time $2^{\mathcal{O}(\mathbf{tw})} \cdot |V(G)|$, improving upon the running time $2^{\mathcal{O}(\mathbf{tw}^2)} \cdot |V(G)|^{\mathcal{O}(1)}$ by Jansen, de Kroon, and Włodarczyk (STOC'21). When a tree decomposition of width $\mathbf{tw}$ is given, then the base of the exponent equals $2^{\omega-1} \cdot 3 + 1$. Our algorithm is based on a novel link between chordal graphs and graphic matroids, which allows us to employ the framework of representative families. On the other hand, we prove that the known $2^{\mathcal{O}(\mathbf{tw} \log \mathbf{tw})} \cdot |V(G)|$-time algorithm for INTERVAL VERTEX DELETION cannot be improved assuming Exponential Time Hypothesis.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases fixed-parameter tractability, treewidth, chordal graphs, interval graphs, matroids, representative families

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.106

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2305.03440`

Funding Supported by the European Research Council (ERC) grant titled PARAPATH. Part of the work has been carried out when the author was a postdoctorate student at the Eindhoven University of Technology.

1 Introduction

Treewidth [32, §7] is arguably the most extensively studied width measure in the graph theory. Simply speaking, treewidth measures to what extent a graph is similar to a tree, where trees and forests are exactly the graphs of treewidth 1. It plays a crucial role in Robertson and Seymour's Graph Minors series [62]. The usefulness of treewidth stems from the fact that a broad class of problems can be solved in linear time on graphs of bounded treewidth. The celebrated Courcelle's Theorem [30] states that any graph problem expressible in the Counting Monadic Second Order Logic (CMSO) can be solved in time $f(\mathbf{tw}) \cdot |V(G)|$, where $\mathbf{tw}$ denotes the treewidth of graph G and f is some computable function. In other words, every such problem is fixed-parameter tractable (FPT) when parameterized by treewidth. Furthermore, bounded-treewidth graphs appear in a wide variety of contexts, which makes treewidth-based algorithms a ubiquitous tool in algorithm design [36, 47, 56, 57, 61]. The function f from Courcelle's Theorem may grow very rapidly and a large body of research has been devoted to optimize the dependency on $\mathbf{tw}$ for particular problems. In the ideal scenario, we would like the function f to be single-exponential, i.e., $f(\mathbf{tw}) = 2^{\mathcal{O}(\mathbf{tw})}$, while possibly allowing a higher (yet constant) exponent at $|V(G)|$. This is often the best we can hope for because sub-exponential running times usually contradict the Exponential Time Hypothesis[1] (ETH) [42].

[1] The Exponential Time Hypothesis states that there exists a constant $\delta > 0$ so that 3-SAT cannot be solved in time $\mathcal{O}(2^{\delta n})$ on n-variable formulas.

50th International Colloquium on Automata, Languages, and Programming (ICALP 2023).
Editors: Kousha Etessami, Uriel Feige, and Gabriele Puppis; Article No. 106; pp. 106:1–106:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

While the standard dynamic programming technique yields single-exponential algorithms for problems with "local constraints", such as Vertex Cover, Dominating Set, or Bipartization, it falls short for problems with "connectivity constraints", such as Feedback Vertex Set, Hamiltonian Cycle, or Connected Vertex Cover, leading to parameter dependency $f(\mathbf{tw}) = 2^{\mathcal{O}(\mathbf{tw} \log \mathbf{tw})}$. On the one hand, this issue was dealt with in the landmark work of Cygan et al. [35], who introduced the Cut & Count technique and obtained randomized single-exponential algorithms for the problems above, among others (see also [59]). In following works, Bodlaender et al. [19] and Fomin et al. [38] presented alternative techniques that allow to circumvent randomization: matrix-based approaches and representative families. On the other hand, Lokshtanov et al. [54] provided a framework for proving "slightly super-exponential" lower bounds under ETH, which paved the way for establishing tight lower bounds for problems that require dependency $f(\mathbf{tw}) = 2^{\mathcal{O}(\mathbf{tw} \log \mathbf{tw})}$. In the same work, they obtained such bounds for Disjoint Paths and Chromatic Number. For problems with a single-exponential dependency $f(\mathbf{tw}) = \mathcal{O}(c^{\mathbf{tw}})$, further research has been devoted to establish the optimal base of the exponent c [31, 33, 35, 53, 67].

Vertex-deletion problems. Many optimization graph problems can be phrased in terms of $\mathcal{H}$-Vertex Deletion: remove the smallest number of vertices from a graph so that the resulting graph belongs to the graph class $\mathcal{H}$. For example, Vertex Cover corresponds to the class $\mathcal{H}$ of edge-less graphs. There is a diverse complexity landscape of ETH-tight running times for various vertex-deletion problems under treewidth parameterization. The classes $\mathcal{H}$ for which tight bounds have been established include: edge-less graphs [53], forests [35] (see also [15]), planar graphs [47, 58], classes defined by a connected forbidden minor [9] (see also [10, 11, 12]), bipartite graphs [53], DAGs [22], even-cycle-free graphs [15, 44], and some classes defined by a forbidden (induced) subgraph [34, 65].

We extend this list by studying the vertex-deletion problems into the classes of chordal and interval graphs. A graph is chordal if it does not contain an induced cycle of length at least 4 (a hole) and a graph is interval if it is an intersection graph of intervals on the real line. Any interval graph is chordal and any chordal graph is perfect. Applications of these two graph classes have been long studied in miscellaneous areas of discrete optimization [8, 14, 25, 50, 60, 63]. On the theoretical side, the treewidth (resp. pathwidth) of a graph G equals the minimum clique number of a chordal (resp. interval) supergraph of G [32, 52]. Moreover, some hard problems become tractable on chordal or interval graphs (or even on graphs with small vertex-deletion distance to chordality) [26, 43, 49].

Our results. The state of the art for Chordal Vertex Deletion (ChVD) is the running time $2^{\mathcal{O}(\mathbf{tw}^2)} n^{\mathcal{O}(1)}$, which follows from a more general result for a hybrid graph measure $\mathcal{H}$-treewidth, where $\mathcal{H} = \texttt{chordal}$ [45]. We improve the dependency on treewidth to single-exponential.

▶ **Theorem 1.1.** *Chordal Vertex Deletion can be solved in deterministic time $\mathcal{O}(c^k k^{\omega+1} n)$ on n-vertex node-weighted graphs when a tree decomposition of width k is provided. The constant c equals $2^{\omega-1} \cdot 3 + 1$.*

Here, $\omega < 2.373$ stands for the matrix multiplication exponent [7]. To prove Theorem 1.1 we establish a new link between chordal graphs and graphic matroids, which allows us to exploit the framework of representative families [37, 38]. ChVD is at least as hard as Feedback Vertex Set, what implies barriers for a significant improvement in the constant c (see Lemma 4.1 and the discussion therein). Thanks to a single-exponential constant-factor FPT approximation for treewidth [20], Theorem 1.1 gives running time $2^{\mathcal{O}(\mathbf{tw})} n$ even when no tree decomposition is provided in the input.

The best known running time for Interval Vertex Deletion is $2^{\mathcal{O}(\mathbf{tw}\log\mathbf{tw})}n$ [64]. (While this algorithm has been described for the edge-deletion variant, we briefly explain in the full version of the article how it can be adapted for vertex deletion.) We show that, unlike the chordal case, this running time is optimal under ETH. This gives a sharp separation between the two studied problems.

▶ **Theorem 1.2.** *Under the assumption of ETH, Interval Vertex Deletion cannot be solved in time $2^{o(\mathbf{tw}\log\mathbf{tw})}n^{\mathcal{O}(1)}$ on n-vertex unweighted graphs of treewidth $\mathbf{tw}$.*

In fact, we show a stronger lower bound that rules out the same running time with respect to a different graph parameter, called treedepth, which is never smaller than treewidth. Our lower bound is obtained via a reduction from $k \times k$ Permutation Clique [54], which produces an instance of size $2^{\mathcal{O}(k)}$ and treedepth $\mathcal{O}(k)$.

Related work. The two considered $\mathcal{H}$-Vertex Deletion problems have been studied in several contexts. Both problems are FPT parameterized by the solution size k, with the best-known running times $\mathcal{O}(8^k(n+m))$ for $\mathcal{H} = \texttt{interval}$ [27] and $2^{\mathcal{O}(k\log k)}n^{\mathcal{O}(1)}$ for $\mathcal{H} = \texttt{chordal}$ [28] (but the problem becomes W[2]-hard for $\mathcal{H} = \texttt{perfect}$ [41]). There are polynomial-time approximation algorithms with approximation factor 8 for $\mathcal{H} = \texttt{interval}$ [27] and $k^{\mathcal{O}(1)}$ for $\mathcal{H} = \texttt{chordal}$ [48]. Observe that, in these two regimes, vertex deletion into chordal graphs seems harder than into interval graphs (although no lower bounds are known to justify such a separation formally); this contrasts our results with respect to the treewidth parameterization.

Both studied problems admit exact exponential algorithms with running times of the form $\mathcal{O}((2-\varepsilon)^n)$ [18] as well as polynomial kernelizations [3, 4, 48]. The obstructions to being chordal (resp. interval) enjoy the Erdős-Pósa property: any graph G either contains k vertex-disjoint subgraphs which are not chordal (resp. not interval) or a vertex set X of size $\mathcal{O}(k^2\log k)$ such that $G - X$ is chordal [51] (resp. interval [2]). Vertex deletion into other subclasses of perfect graphs has been studied as well [1, 5, 6, 68]. For other modification variants, where instead of vertex deletions one considers removals, insertions, or contractions of edges, see, e.g., [17, 26, 27, 28, 39, 55, 70].

The concept of representative families, which plays an important role in our algorithm for ChVD, has found applications outside the context of treewidth as well [66, 71]. Our other tool, boundaried graphs, has revealed fruitful insights for various graph classes [9, 21, 45].

Organization of the paper. We begin by describing our technical contributions informally in Section 2. We provide basic preliminaries in Section 3, while the extended preliminaries including tree decompositions and representative families can be found in the full version of the article. Section 4 is devoted to establishing a connection between chordal graphs and graphic matroids. The description of the dynamic programming algorithm over a tree decomposition follows standard conventions and is provided in the full version. In Section 5 we prove our lower bound for Interval Vertex Deletion. We conclude in Section 6. The proofs of statements indicated with ($\star$) are postponed to the full version. The numbering of statements is adjusted to match in both versions.

2 Techniques

Chordal Vertex Deletion. The standard approach to design algorithms over a bounded-width tree decomposition is to assign a data structure to each node t in the decomposition, which stores information about partial solutions for the subgraph associated with the subtree of t. Suppose that $X \subseteq V(G)$ is a bag of t, $A \subseteq V(G) \setminus X$ denote the set of vertices appearing in the bags of the descendants of t (but not in X), and $B \subseteq V(G)$ is the set of remaining vertices. We say that a subset $S \subseteq V(G)$ is a *solution* if $G[S]$ is chordal; we want to *maximize* the size of S. Next, a pair $(S_A \subseteq A, S_X \subseteq X)$ is a *partial solution* if $G[S_A \cup S_X]$ is chordal. A set $S_B \subseteq B$ is an *extension* of a partial solution (S_A, S_X) if $S_A \cup S_X \cup S_B$ is a solution. Since X separates S_A from S_B, the graph $G[S_A \cup S_X \cup S_B]$ can be regarded as a result of *gluing* $G[S_A \cup S_X]$ with $G[S_B \cup S_X]$ alongside the *boundary* S_X. For a node t and $S_X \subseteq X$, we want to store a family of partial solutions $\mathcal{G}_{t,S_X}$ so that for every possible $S_B \subseteq B$: if S_B is an extension for some partial solution (S_A, S_X), then there exists a partial solution $(S'_A, S_X) \in \mathcal{G}_{t,S_X}$ for which (a) S_B is still a valid extension, and (b) S'_A is at least as large as S_A. We say that such a family satisfies the correctness invariant for (t, S_X).

Jansen et al. [45] showed that any chordal graph H with a boundary of size k can be *condensed* to a graph H' on $\mathcal{O}(k)$ vertices that exhibits the same behavior in terms of gluing. More precisely, the gluing product of H with any graph J is chordal if and only if the gluing product of H' with J is chordal. Since there are $2^{\mathcal{O}(\mathbf{tw}^2)}$ graphs on $\mathcal{O}(\mathbf{tw})$ vertices and $2^{\mathcal{O}(\mathbf{tw})}$ choices for the boundary S_X, it suffices to store only $2^{\mathcal{O}(\mathbf{tw}^2)}$ partial solutions.

We take this idea one step further and show that it is actually sufficient to store only $2^{\mathcal{O}(\mathbf{tw})}$ partial solutions. To this end, we investigate the properties of the class of chordal graphs with respect to the gluing operation and prove a homomorphism theorem relating it to graphic matroids. A *graphic matroid* of a graph J is a set system $\mathcal{I}$ over $E(J)$ where a subset $S \subseteq E(J)$ belongs to $\mathcal{I}$ (and is called *independent*) when S contains no cycles. A *rank* of a matroid is the largest size of an independent set; here this coincides with the size of any spanning forest in J. In the following statement, $\mathcal{G}_{X,B}$ is a family of graphs H that satisfy (a) $V(H) \supseteq X$ and (b) $H[X] = B$. For graphs $H_1, H_2 \in \mathcal{G}_{X,B}$ we assume that $V(H_1) \cap V(H_2) = X$ and define their gluing product as $H_3 = (H_1, X) \oplus (H_2, X)$ where $V(H_3) = V(H_1) \cup V(H_2)$ and $E(H_3) = E(H_1) \cup E(H_2)$.

▶ **Theorem 2.1.** *Consider a family of graphs $\mathcal{G}_{X,B}$ for some pair (X, B). There exists a graphic matroid $M = (E, \mathcal{I})$ of rank at most $|X| - 1$ and a polynomial-time computable mapping $\sigma : \mathcal{G}_{X,B} \to 2^E$ such that $(H_1, X) \oplus (H_2, X)$ is chordal if and only if $\sigma(H_1) \cap \sigma(H_2) = \emptyset$ and $\sigma(H_1) \cup \sigma(H_2) \in \mathcal{I}$.*

With this criterion at hand, we can employ the machinery of representative families to truncate the number of partial solutions to be stored for a node of a tree decomposition. Technical details aside, for a family $\mathcal{S}$ of independent sets in a matroid $M = (E, \mathcal{I})$, a subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ is called *representative* for $\mathcal{S}$ if for every independent set Y in M: if there exists $X \in \mathcal{S}$ so that $X \cap Y = \emptyset$ and $X \cup Y \in \mathcal{I}$, then there exists $\widehat{X} \in \widehat{\mathcal{S}}$ so that $\widehat{X} \cap Y = \emptyset$ and $\widehat{X} \cup Y \in \mathcal{I}$. Fomin et al. [38] showed that for any family $\mathcal{S}$ in a graphic matroid (more generally, in a linear matroid) of rank k there exists a representative family of size at most 2^k and it can be constructed in time $2^{\mathcal{O}(k)}$. We use Theorem 2.1 to translate this result into the language of chordal graphs and gluing. When $\mathcal{G}_{t,S_X}$ is a family of partial solutions that satisfies the correctness invariant for (t, S_X), a representative family for $\sigma(\mathcal{G}_{t,S_X})$ in the related graphic matroid M corresponds to a subfamily $\widehat{\mathcal{G}}_{t,S_X} \subseteq \mathcal{G}_{t,S_X}$ that satisfies condition (a) of the correctness invariant and $|\widehat{\mathcal{G}}_{t,S_X}| \le 2^{\mathbf{tw}}$. In order to satisfy condition (b), we need to assign weights to the elements of the matroid M, encoding the size of the largest partial solution mapped to each element. We can then utilize the weighted variant of representative families, which preserves the largest-weight elements [38]. By storing only the condensed forms of the partial solutions (having $\mathcal{O}(\mathbf{tw})$ vertices), we also achieve a linear dependency on $|V(G)|$.

In order to prove Theorem 2.1, we give a novel criterion for testing chordality of a gluing product. When G originates from gluing two chordal graphs G_1, G_2 alongside boundary X, then any hole in G must visit both $V(G_1) \setminus X$ and $V(G_2) \setminus X$, so it must traverse X multiple times. We show that if a hole H intersects at least two connected components of

$G[X]$, then it corresponds to a cycle in the graph obtained from G by contracting each of the connected components of $G[X]$, $G_1 - X$, $G_2 - X$ into single vertices. Otherwise, let C be the unique connected component of $G[X]$ that is intersected by the hole. We prove that there exists a vertex set $S \subseteq V(C)$ that is disjoint from $V(H)$ and $C - S$ has two connected components C_1, C_2 satisfying $N_C(C_1) = N_C(C_2) = S$ (below we refer to such components as *relevant*) and having non-empty intersections with $V(H)$. Moreover, every vertex from $V(H) \cap C$ belongs to some relevant component. Consider a graph $\texttt{Aux}(G, X, S)$ obtained from G by (1) removing the connected components of $G[X]$ different than C, (2) contracting relevant components of $C - S$ into single vertices while removing the irrelevant ones, and (3) contracting the components of $G_1 - X$, $G_2 - X$ into single vertices. A detailed construction is given in Definition 4.10; see also Figure 1 on page 9. Then the hole H corresponds to a cycle in $\texttt{Aux}(G, X, S)$. The first scenario can be analyzed with this approach as well, by taking $S = \emptyset$. We prove that considering all minimal vertex separators S in $G[X]$ and checking acyclity of each auxiliary graph $\texttt{Aux}(G, X, S)$ yields a necessary and sufficient condition for G to be chordal.

This criterion allows us to construct a graphic matroid encoding all the information about each of the graphs G_1, G_2 necessary to reconstruct the graphs $\texttt{Aux}(G, X, S)$ and to determine whether G is chordal. In order to bound the rank of this matroid, we investigate the structure of minimal vertex separators in a chordal graph and bound the size of a spanning forest in a certain graph obtained from the union of $\texttt{Aux}(G, X, S)$. A criterion of a similar kind is known for testing planarity of a gluing product of planar graphs when the boundary has a Hamiltonian cycle; then the corresponding auxiliary graph (defined in a different way) should be bipartite [13]. Our criterion can be also compared to the one used by Bonnet et al. [23] for analyzing gluing products with respect to certain subclasses of chordal graphs. We elaborate more on their approach in the full version of the paper.

Interval Vertex Deletion. In order to prove Theorem 1.2 we present a parameterized reduction from $k \times k$ Permutation Clique. Here, the input is a graph G on vertex set $[k] \times [k]$, and we ask whether there exists a permutation $\pi\colon [k] \to [k]$ such that $(1, \pi(1)), (2, \pi(2)), \dots, (k, \pi(k))$ forms a clique in G. Lokshtanov et al. [54] proved that $k \times k$ Permutation Clique cannot be solved in time $2^{o(k \log k)}$ under ETH. So we seek a reduction from $k \times k$ Permutation Clique to Interval Vertex Deletion that produces a graph of treewidth $\mathcal{O}(k)$.

Imagine an interval model of a complete graph Y on vertex set $[k]$ in which all the right endpoints of the intervals coincide and all the left endpoints are distinct. Choosing the order of the left endpoints encodes some permutation $\pi\colon [k] \to [k]$ (see Figure 2 on page 13). We can extend this interval model by inserting a new vertex v only if $N(v)$ corresponds to a set of intervals intersecting at a single point. This is possible only when $N(v) = \pi([\ell])$ for some $\ell \in [k]$. Furthermore, inserting to Y independent vertices $v_1, v_2, \dots, v_k$, such that $|N(v_i)| = i$ and $N(v_i) \subset N(v_{i+1})$, enforces the choice of permutation π. We can thus encode a permutation π by an ascending family of sets $N_1 \subset N_2 \subset \dots \subset N_k = [k]$, satisfying $N_i = \pi([i])$, which correspond to the neighborhoods of $v_1, v_2, \dots, v_k$ in Y. On the other hand, any ascending family of sets for which the construction above gives an interval graph, must encode some permutation. On an intuitive level, a partial interval model of a size-k separator can encode one of $k!$ permutations.

We need a mechanism to verify that a chosen permutation π encodes a clique, i.e., that it satisfies $\binom{k}{2}$ constraints of the form $(i, \pi(i))(j, \pi(j)) \in E(G)$. To implement a single constraint, we construct a *choice gadget*, inspired by the reduction to Planar Vertex Deletion [58]. Such a gadget $C_{i,j}$ is defined as a path-like structure, divided into blocks, so that each block has some special vertices adjacent to Y (see Figure 3 on page 14). We show that any minimum-size interval deletion set in $C_{i,j}$ must "choose" one block and leave its special vertices untouched while it can remove the remaining special vertices. We use this gadget to check if a permutation π encoded by an ascending family of sets $N_1 \subset N_2 \subset \cdots \subset N_k$ satisfies the constraint $(i,\pi(i))(j,\pi(j)) \in E(G)$. As $\pi(i)$ is the only element in $N_i \setminus N_{i-1}$, this information can be extracted from the tuple $(N_{i-1}, N_i, N_{j-1}, N_j)$. We create a single block in $C_{i,j}$ for each valid tuple. Since the number of such tuples is $2^{\mathcal{O}(k)}$, we need a choice gadget of exponential length, unlike the mentioned reduction which works in polynomial time. However, producing an instance of size $2^{\mathcal{O}(k)}$ and treewidth $\mathcal{O}(k)$ is still sufficient to achieve the claimed lower bound.

3 Preliminaries

We write $[k] = \{1, 2, \dots, k\}$ and assume that $[0] = \emptyset$. We abbreviate $X \setminus v = X \setminus \{v\}$. For a function $w \colon X \to \mathbb{N}$ and $S \subseteq X$ we use shorthand $w(S) = \sum_{x \in S} w(x)$. We follow the standard notational conventions for graphs, which are omitted from this extended abstract.

Separators. For vertices $u, v \in V(G)$ a vertex set $S \subseteq V(G) \setminus \{u, v\}$ is called a (u,v)-separator if u, v belong to different connected components of $G - S$. A (u,v)-separator is minimal when no proper subset of it is a (u,v)-separator. A vertex set S is called a minimal vertex separator if S is a minimal (u,v)-separator for some $u, v \in V(G)$.

▶ **Lemma 3.1** ($\star$). *Let u, v be vertices in a graph G and S be a (u,v)-separator in G. Denote by C_u, C_v the connected components of $G - S$ that contain respectively u and v. Then S is minimal if and only if $N_G(C_u) = N_G(C_v) = S$.*

A vertex (or a vertex set) is called *simplicial* if its open neighborhood is a clique.

▶ **Lemma 3.2** ($\star$). *Let S be a minimal vertex separator in a graph G. Then S does not contain any simplicial vertices.*

Chordal and interval graphs. An interval graph is an intersection graph of intervals on the real line. In an interval model $\mathcal{I}_G = \{I(v) \mid v \in V(G)\}$ of a graph G, each vertex $v \in V(G)$ corresponds to a closed interval $I(v)$; there is an edge between vertices u and v if and only if $I(v) \cap I(u) \neq \emptyset$.

A *hole* in a graph is an induced (i.e., chordless) cycle of length at least four. A graph is chordal when it does not contain any hole. An equivalent definition states that a chordal graph is an intersection graph of a family of subtrees in a tree [40]. This implies that any interval graph is chordal. For more background on these graph classes see surveys [16, 24].

The characterization of the two classes as intersection graphs of intervals/subtrees leads to the following observation.

▶ **Observation 3.3.** *The classes of chordal and interval graphs are closed under vertex deletions and edge contractions.*

An *asteroidal triple* (AT) is a triple of vertices such that for any two of them there exists a path between them avoiding the closed neighborhood of the third. Interval graphs cannot contain ATs, which is a consequence of a linear ordering of any interval model. It turns out that this is the only property that separates the two graph classes.

▶ **Lemma 3.4** ([24]). *A graph is interval if and only if it is chordal and does not contain an AT.*

We collect two more useful facts about chordal graphs.

▶ **Lemma 3.5** ([24]). *Every non-empty chordal graph contains a simplicial vertex.*

When a chordal graph contains a cycle then it also contains a triangle. As a bipartite graph does not have any triangles, we obtain the following.

▶ **Observation 3.6.** *If a graph is chordal and bipartite, then it is a forest.*

A vertex set S in graph G is called a *chordal deletion set* (resp. *interval deletion set*) if $G - S$ is chordal (resp. interval). The Chordal/Interval Vertex Deletion problem is defined as follows. We are given a graph G, a non-negative weight function $w\colon V(G) \to \mathbb{N}$, an integer p, and we ask whether there exists a chordal (resp. interval) deletion set S in G such that $w(S) \le p$.

Boundaried graphs. For a set X and a graph B on vertex set X, we define a family $\mathcal{G}_{X,B}$ of graphs G that satisfy (a) $V(G) \supseteq X$, (b) $G[X] = B$. For graphs $G_1, G_2 \in \mathcal{G}_{X,B}$ we define their gluing product $(G_1, X) \oplus (G_2, X)$ by taking a disjoint union of G_1 and G_2 and identifying vertices from X. Note that two vertices from X are adjacent in G_1 if and only if they are adjacent in G_2.

For $X \subseteq V(G)$ a pair (G, X) is called a boundaried graph. We say that two boundaried graphs $(G_1, X), (G_2, X)$ are compatible if $G_1, G_2 \in \mathcal{G}_{X,B}$ for some B. We remark that it is common in the literature to define a boundaried graph as a triple (G, X, λ) where $\lambda\colon X \to [|X|]$ is a labeling (cf. [9, 21]). Since we do not need to perform gluing of abstract boundaried graphs, but only ones originating from subgraphs of a fixed graph, this simpler definition is sufficient.

As an example, consider a graph G and $X \subseteq V(G)$. Then for any $A \subseteq V(G) \setminus X$ the graph $G[A \cup X]$ belongs to $\mathcal{G}_{X,G[X]}$. When $A, B \subseteq V(G) \setminus X$ are disjoint and non-adjacent then $G[A \cup B \cup X]$ is isomorphic to $(G[A \cup X], X) \oplus (G[B \cup X], X)$.

4 Chordal Deletion

We begin with a simple treewidth-preserving reduction from Feedback Vertex Set.

▶ **Lemma 4.1** ($\star$). *Let G be a graph and $\ell \in \mathbb{N}$. Let G' be obtained from G by subdividing each edge. Then $\textbf{tw}(G') = \textbf{tw}(G)$ and G has a feedback vertex set (FVS) of size ℓ if and only if G' has a chordal deletion set of size ℓ.*

As a consequence, the base of the exponent c in Theorem 1.1 must be at least 3 under Strong Exponential Time Hypothesis [35] and c must be at least $2^\omega + 1$ if the current-best deterministic algorithm for Feedback Vertex Set parameterized by treewidth is optimal [69]. While we have no evidence that the mentioned algorithm should be optimal for deterministic time, we provide this comparison to indicate that breaching this gap for ChVD would imply the same for a more heavily studied problem.

Minimal vertex separators. We set the stage for the proof of Theorem 2.1. First we need to develop some theory about minimal vertex separators in chordal graphs.

▶ **Definition 4.2.** *Let $\texttt{MinSep}(G)$ denote the set of minimal vertex separators in a graph G. For a graph G and a (possibly empty) set $S \subseteq V(G)$, we define $\texttt{Comp}(G, S)$ to be the set of connected components C_i of $G - S$ for which it holds that $N_G(C_i) = S$.*

Note that whenever G is disconnected then $\emptyset \in \texttt{MinSep}(G)$ and $\texttt{Comp}(G, \emptyset)$ is just the set of connected components of G. According to Lemma 3.1, the set S is a minimal (u, v)-separator if and only if u, v belong to some (distinct) components from $\texttt{Comp}(G, S)$. For later use, we establish a relation between sets $\texttt{MinSep}(G)$, $\texttt{Comp}(G, S)$ in G and a graph obtained by a removal of a simplicial vertex.

▶ **Lemma 4.3** ($\star$). *Let v be a simplicial vertex in G and $S \in \mathtt{MinSep}(G)$. If $S \neq N_G(v)$ then $S \in \mathtt{MinSep}(G - v)$ and $|\mathtt{Comp}(G, S)| = |\mathtt{Comp}(G - v, S)|$.*

We need a simple technical lemma about minimal vertex separators.

▶ **Lemma 4.4** ($\star$). *Let G be a connected graph and $V_1, \dots, V_k \subseteq V(G)$, $k \geq 2$, be disjoint sets so that $G[V_i]$ is connected, for $i \in [k]$, and $E_G(V_i, V_j) = \emptyset$, for $i \neq j$. Then there exists a minimal vertex separator $S \subseteq V(G) \setminus (V_1 \cup \dots \cup V_k)$ in G which is a (V_i, V_j)-separator for some $i \neq j$ and each set V_i is contained in some component $C \in \mathtt{Comp}(G, S)$.*

We will use the following concept which appears in the current-best algorithm for ChVD by Jansen et al [45]. In the full version, we also provide several properties of this operation, used to process partial solutions in a treewidth DP.

▶ **Definition 4.5** ([46, Def. 5.55]). *For a graph G and a vertex set $X \subseteq V(G)$ let the graph $\mathtt{Condense}(G, X)$ be obtained from G by contracting the connected connected components of $G - X$ into single vertices and then removing those of them which are simplicial.*

In this section we will exploit the following property of condensation.

▶ **Lemma 4.9** ($\star$). *Consider a graph G with a vertex set X so that $G[X]$ is chordal. Then G is chordal if and only if the following conditions hold:*
1. *for each connected component C of $G - X$ the graph $G[X \cup C]$ is chordal,*
2. *the graph $\mathtt{Condense}(G, X)$ is chordal.*

In order to turn Lemma 4.9 into a more convenient criterion, we will compress information about a graph G with a vertex subset X into multiple auxiliary graphs, one for each minimal vertex separator in $G[X]$.

▶ **Definition 4.10.** *Consider a graph G with a vertex set X so that $G[X]$ is chordal. For a set $S \in \mathtt{MinSep}(G[X])$ we construct the graph $\mathtt{Aux}(G, X, S)$ as follows:*
1. *contract each $C \in \mathtt{Comp}(G[X], S)$ into a vertex and remove the remaining vertices of X (including all of S),*
2. *contract each connected component of $G - X$ into a vertex.*

Note that $\mathtt{Aux}(G, X, \emptyset)$ is obtained by just contracting each connected component of $G[X]$ and each connected component of $G - X$. Moreover, observe that $\mathtt{Aux}(G, X, S)$ is always a bipartite graph because there can be no edges between two components from $\mathtt{Comp}(G[X], S)$ nor between two components of $G - X$. See Figure 1 for an example of this construction.

To make a connection between holes in G and cycles in $\mathtt{Aux}(G, X, S)$, we need a criterion to derive existence of a cycle from a closed walk with certain properties. In the following lemma we consider a cyclic order on a sequence of length k. We define the successor operator as $s(i) = i + 1$, for $i \in [k - 1]$, and $s(k) = 1$.

▶ **Lemma 4.11** ($\star$). *Let G be a bipartite graph with vertex partition $V(G) = A \cup B$. Suppose there exists a sequence of vertices $(v_1, \dots, v_k)$ in G such that:*
1. *for $i \in [k]$ it holds $v_i = v_{s(i)}$ or $v_i v_{s(i)} \in E(G)$,*
2. *the multiset $\{v_1, \dots, v_k\}$ contains at most one occurrence of each vertex from A,*
3. *the set $\{v_1, \dots, v_k\}$ contains at least two vertices from B.*

Then G contains a cycle.

We are ready to prove a proposition creating a link between chordality and acyclicity.

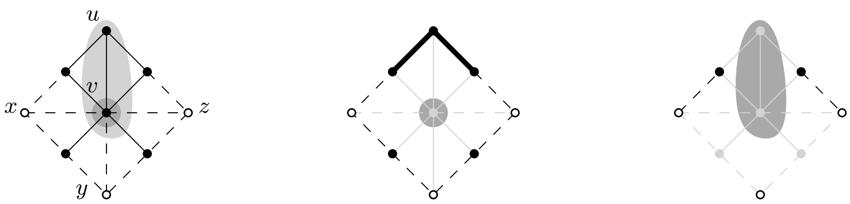

Figure 1 On the left: graph G and set $X \subseteq V(G)$ represented by black disks. The graph $G[X]$ is drawn with solid edges. There are two minimal vertex separators in $G[X]$: $S_1 = \{v\}$ and $S_2 = \{u, v\}$, sketched in gray. In the middle: the graph $\texttt{Aux}(G, X, S_1)$ with thick edges indicating a component that gets contracted into a single vertex; the gray vertices and edges are removed. On the right: the graph $\texttt{Aux}(G, X, S_2)$; note that $|\texttt{Comp}(G[X], S_2)| = 2$ because the lower vertices of X are not adjacent to every vertex in S_2. The graph $\texttt{Aux}(G, X, S_1)$ contains a cycle and this witnesses that G is not chordal. However, removing from G any single vertex among x, y, z results in a chordal graph.

▶ **Proposition 4.12.** *Consider a graph G with a vertex subset $X \subseteq V(G)$ so that for each connected component C of $G - X$ the graph $G[X \cup C]$ is chordal. Then G is chordal if and only if for each $S \in \texttt{MinSep}(G[X])$ the graph $\texttt{Aux}(G, X, S)$ is acyclic.*

Proof. First we argue that if G is chordal then all graphs $\texttt{Aux}(G, X, S)$ are acyclic. Because the class of chordal graphs is closed under vertex deletions and edge contractions, the graphs $\texttt{Aux}(G, X, S)$ are chordal as well. Since each graph $\texttt{Aux}(G, X, S)$ is also bipartite, by Observation 3.6 we obtain that $\texttt{Aux}(G, X, S)$ is acyclic.

Now suppose that G is not chordal. Let $G' = \texttt{Condense}(G, X)$ (recall Definition 4.5). By Lemma 4.9, the graph G' is not chordal as well but for each vertex $v \in V(G') \setminus X$ the graph $G'[X \cup \{v\}]$ is chordal (because contraction preserves chordality). Note that $\texttt{Aux}(G', X, S)$ is an induced subgraph of $\texttt{Aux}(G, X, S)$ for each $S \in \texttt{MinSep}(G[X])$ (they may differ only due to removal of simplicial vertices), so it suffices to show that one of the graphs $\texttt{Aux}(G', X, S)$ has a cycle.

As G' is not chordal, it contains a hole $H = (u_1, \ldots, u_k)$. We consider two cases: either $V(H)$ intersects at least two connected components of $G'[X]$ or only one. In the first case, let $\phi_0 \colon V(G') \to V(\texttt{Aux}(G', X, \emptyset))$ be the mapping given by the contractions from Definition 4.10. Recall that $V(G') \setminus X$ is an independent set in G' so ϕ_0 is an identity on this set. The sequence $(\phi_0(u_1), \ldots, \phi_0(u_k))$ meets the preconditions of Lemma 4.11 for $A = V(G') \setminus X$ and $B = \phi_0(X)$ so $\texttt{Aux}(G', X, \emptyset)$ has a cycle. As $G'[X] = G[X]$ is disconnected, we have $\emptyset \in \texttt{MinSep}(G[X])$.

In the second case, let $Y \subseteq X$ induce the only connected component of $G'[X]$ that intersects $V(H)$. Let $V_1, \ldots, V_\ell \subseteq Y$ be the vertex sets of maximal subpaths of H within Y. By the definition of a hole, we have $E_{G'}(V_i, V_j) = \emptyset$ for distinct $i, j \in [\ell]$. It must be $\ell \geq 2$ because for each $v \in V(G') \setminus X$ the graph $G'[X \cup \{v\}]$ is chordal and the hole H must visit at least two vertices from the independent set $V(G') \setminus X$. By Lemma 4.4, there exists a minimal vertex separator $S \subseteq Y \setminus V(H)$ in $G'[Y]$ such that every set V_i is contained in some component from $\texttt{Comp}(G'[Y], S)$ and at least two components from $\texttt{Comp}(G'[Y], S)$ intersect $V(H)$. Note that $S \in \texttt{MinSep}(G[X])$. Let C_S be the union of the components from $\texttt{Comp}(G'[Y], S)$; note that $V(H) \subseteq V(C_S) \cup (V(G') \setminus X)$.

Let $\phi_S \colon V(C_S) \cup (V(G') \setminus X) \to V(\texttt{Aux}(G', X, S))$ be the mapping given by the contractions from Definition 4.10 which turn each component from $\texttt{Comp}(G'[Y], S)$ into a single vertex. Again, the sequence $(\phi_S(u_1), \ldots, \phi_S(u_k))$ meets the preconditions of Lemma 4.11 for $A = V(G') \setminus X$ and $B = \phi_S(V(C_S))$ so $\texttt{Aux}(G', X, S)$ has a cycle. See Figure 1 for an illustration. ◀

Signatures of boundaried graphs. The next step is to construct a graphic matroid M_B for a chordal graph B so that for any two graphs $G_1, G_2 \in \mathcal{G}_{X,B}$ the information about chordality of $(G_1, X) \oplus (G_2, X)$ could be read from M_B. Proposition 4.12 already relates chordality to acyclicity but the corresponding graphic matroids for G_1, G_2 are disparate. To circumvent this, we will further compress the information about cycles.

▶ **Definition 4.13.** *Consider a graph B. For $S \in \mathtt{MinSep}(B)$, let $\mathtt{Base}(B, S)$ be the complete graph on vertex set $\mathtt{Comp}(B, S)$. The graph $\mathtt{Base}(B)$ is a disjoint union of all the graphs $\mathtt{Base}(B, S)$ for $S \in \mathtt{MinSep}(B)$.*

That is, we treat the components from $\mathtt{Comp}(B, S)$ as abstract vertices of a new graph which is a union of cliques.

The following transformation is similar to the one used in the algorithm for Steiner Tree based on representative families [38]. For the sake of disambiguation, in the definition below we assume an implicit linear order on the vertices of B; this order may be arbitrary. Since vertices of $\mathtt{Base}(B)$ correspond to distinct subsets of $V(B)$, which can ordered lexicographically, fixing the order on $V(B)$ yields an order on $V(\mathtt{Base}(B))$. We can thus assume that also the vertices of $V(\mathtt{Base}(B))$ are linearly ordered.

▶ **Definition 4.14.** *Consider a chordal graph B and $Y \subseteq V(B)$. We define the* spanning signature *$\mathtt{Span}(B, Y) \subseteq E(\mathtt{Base}(B))$ as follows. For each $S \in \mathtt{MinSep}(B)$ let $C_{S,Y} \subseteq V(\mathtt{Base}(B, S))$ be given by components from $\mathtt{Comp}(B, S)$ with a non-empty intersection with Y. Let $P_{S,Y} \subseteq E(\mathtt{Base}(B, S))$ be the path connecting the vertices of $C_{S,Y}$ in the increasing order. Then $\mathtt{Span}(B, Y) = \bigcup_{S \in \mathtt{MinSep}(B)} P_{S,Y}$.*

In other words, $\mathtt{Span}(B, Y)$ is a disjoint union of paths in the graph $\mathtt{Base}(B)$, where each path encodes the relation between Y and a respective minimal vertex separator in B.

The next lemma states that under certain conditions replacing a vertex v with a tree over $N(v)$ (in particular: a path) does not affect acyclicity of the graph. Note that due to the precondition $|N(u) \cap N(v)| \le 1$ we never attempt to insert an edge that is already present.

▶ **Lemma 4.15** ($\star$). *Let G be a bipartite graph with a vertex partition $V(G) = A \cup B$ so that for each distinct $u, v \in A$ it holds that $|N_G(u) \cap N_G(v)| \le 1$. Consider a graph G' obtained from G by replacing each vertex $v \in A$ by an arbitrary tree on vertex set $N_G(v)$. Then G is acyclic if and only if G' is acyclic.*

This allows us to translate the criterion from Proposition 4.12 into a more convenient one, in which the vertex set of the auxiliary graph depends only on $G[X]$ rather than G.

▶ **Lemma 4.16.** *Consider a graph G with a vertex subset $X \subseteq V(G)$. Let $\mathcal{C}$ denote the family of connected components of $G - X$. Suppose that for each $C \in \mathcal{C}$ the graph $G[X \cup C]$ is chordal. Then G is chordal if and only if:*

1. *the sets $\mathtt{Span}(G[X], N_G(C))$, for different $C \in \mathcal{C}$, are pairwise disjoint,*
2. *the union of sets $\mathtt{Span}(G[X], N_G(C))$, over $C \in \mathcal{C}$, forms an acyclic edge set in $E(\mathtt{Base}(G[X]))$.*

Proof. From Proposition 4.12 we know that G is chordal if and only if for each $S \in \mathtt{MinSep}(G[X])$ the graph $\mathtt{Aux}(G, X, S)$ is acyclic. We consider two cases.

First, suppose that for some $S \in \mathtt{MinSep}(G[X])$ there are two vertices representing distinct components $C_1, C_2 \in \mathcal{C}$ that share two common neighbors x, y in $\mathtt{Aux}(G, X, S)$. In other words, there are two components from $\mathtt{Comp}(G[X], S)$ that intersect both $N_G(C_1)$ and $N_G(C_2)$. Then $\mathtt{Aux}(G, X, S)$ contains a cycle of length 4, so G is not chordal. If $\mathtt{Span}(G[X], N_G(C_1))$

and $\texttt{Span}(G[X], N_G(C_2))$ share an edge, then condition (1) fails, so suppose this is not the case. But then the paths $P_{S,N(C_1)}$ and $P_{S,N(C_2)}$ (recall Definition 4.14) are edge-disjoint and they both visit x and y. As a consequence, x, y lie on a cycle contained in the edge set $\texttt{Span}(G[X], N_G(C_1)) \cup \texttt{Span}(G[X], N_G(C_2))$ so condition (2) fails. In summary, both G is not chordal and one of conditions (1, 2) does not hold.

Next, suppose that for each $S \in \texttt{MinSep}(G[X])$ and any two vertices representing distinct components $C_1, C_2 \in \mathcal{C}$ the intersection of their neighborhoods in $\texttt{Aux}(G, X, S)$ contains at most one element. This implies condition (1). Consider a graph H given by a disjoint union of all graphs $\texttt{Aux}(G, X, S)$ over $S \in \texttt{MinSep}(G[X])$. This graph meets the preconditions of Lemma 4.15. Replacing each $\mathcal{C}$-component-vertex in $\texttt{Aux}(G, X, S)$ by the path $P_{S,N(C)}$ transforms H into a subgraph of $\texttt{Base}(G[X])$ with the edge set $\bigcup_{C \in \mathcal{C}} \texttt{Span}(G[X], N_G(C))$. By Lemma 4.15, this graph is acyclic if and only if the graph H is. By Proposition 4.12, this condition is equivalent to G being chordal. The lemma follows. ◀

We are ready to define the graphic matroid encoding all the necessary information about where a hole can appear after gluing two chordal graphs. Recall that a graphic matroid of a graph G is a set system over $E(G)$ where a subset $S \subseteq E(G)$ is called independent when S contains no cycles. More information about matroids can be found in the preliminaries of the full version of the article.

▶ **Definition 4.17.** *For a graph B on vertex set X we define matroid M_B as the graphic matroid of the graph $\texttt{Base}(B)$. For a graph $G \in \mathcal{G}_{X,B}$ the signature $\texttt{Sign}(G, X) \subseteq E(\texttt{Base}(B))$ is defined as a union of $\texttt{Span}(B, N_G(C))$ over all connected components C of $G - X$.*

It follows from Lemma 4.16 that whenever G is chordal then $\texttt{Sign}(G, X)$ is acyclic and so it forms an independent set in the matroid $M_{G[X]}$. We can now give the existential part of Theorem 2.1. The mapping $\sigma \colon \mathcal{G}_{X,B} \to 2^{E(M_B)}$ therein is given here as $\sigma(G) = \texttt{Sign}(G, X)$.

▶ **Lemma 4.18** ($\star$). *Let (G_1, X) and (G_2, X) be compatible boundaried chordal graphs. Then $G = (G_1, X) \oplus (G_2, X)$ is chordal if and only if the sets $\texttt{Sign}(G_1, X)$, $\texttt{Sign}(G_2, X) \subseteq E(\texttt{Base}(G[X]))$ are disjoint and $\texttt{Sign}(G_1, X) \cup \texttt{Sign}(G_2, X)$ is acyclic.*

Furthermore, $\texttt{Sign}(G, X) = \texttt{Sign}(G_1, X) \cup \texttt{Sign}(G_2, X)$.

The following lemma is the main ingredient in the running time analysis. As the bound on the representative family's size is exponential in the rank of a matroid[2], it is necessary to bound the rank of M_B. It is known that the number of minimal vertex separators in a chordal graph is bounded by the number of vertices but we need a strengthening of this fact.

▶ **Lemma 4.19.** *For a non-empty chordal graph B, the rank of M_B is at most $|V(B)| - 1$.*

Proof. Let $k = |V(B)|$. The rank of M_B equals the size of a spanning forest in $\texttt{Base}(B)$. The vertex sets of connected components of $\texttt{Base}(B)$ are the sets $\texttt{Comp}(B, S)$ for $S \in \texttt{MinSep}(B)$. Therefore it suffices to estimate

$$\sum_{S \in \texttt{MinSep}(B)} (|\texttt{Comp}(B, S)| - 1) \leq k - 1.$$

We first prove the inequality for connected chordal graphs by induction on k. For $k = 1$ the sum is zero. Consider $k > 1$. By Lemma 3.5, B contains a simplicial vertex. Let v be a simplicial vertex in B and suppose that the claim holds for the graph $B - v$ (which is

[2] We remark that Fomin et al. [38] also considered a case when the rank might be large and the exponential term is governed by a different parameter but it is not applicable in our case.

connected). Let S be a minimal vertex separator in B. By Lemma 4.3 when $S \neq N_B(v)$ then $S \in \texttt{MinSep}(B-v)$ and $|\texttt{Comp}(B,S)| = |\texttt{Comp}(B-v,S)|$. In that case the summand coming from S is the same for B and $B-v$.

It remains to handle the case $S = N_B(v)$. Clearly, $\{v\} \in \texttt{Comp}(B,S)$. If $|\texttt{Comp}(B,S)| = 1$ then $S \notin \texttt{MinSep}(B)$ (Lemma 3.1). If $|\texttt{Comp}(B,S)| = 2$ then $S \in \texttt{MinSep}(B) \setminus \texttt{MinSep}(B-v)$ and the sum grows by one. If $|\texttt{Comp}(B,S)| \geq 3$ then $S \in \texttt{MinSep}(B) \cap \texttt{MinSep}(B-v)$ and $|\texttt{Comp}(B,S)| = |\texttt{Comp}(B-v,S)| + 1$ so the sum again grows by one. This concludes the proof of the inequality for connected chordal graphs.

When B is disconnected, let $B_1, B_2, \dots, B_t$ denote its connected components and let $k_i = |V(B_i)|$. We have $|\texttt{Comp}(B,\emptyset)| - 1 = t - 1$. Together with the sums for $B_1, B_2, \dots, B_t$ the total sum is at most $\sum_{i=1}^{t} k_i - t + t - 1 = k - 1$. ◀

The last thing to be checked is whether we can compute the signatures efficiently. To this end, we enumerate minimal vertex separators using Lemma 4.3.

▶ **Lemma 4.20** (⋆). *There is a polynomial-time algorithm that, given a graph G with a vertex subset $X \subseteq V(G)$ such that $G[X]$ is chordal, computes $\texttt{Sign}(G, X)$.*

Lemmas 4.18, 4.19, and 4.20 entail Theorem 2.1 but instead of working with that abstract statement we use these three lemmas directly when describing the final algorithm. The results of this section allow us to employ the framework of representative families in order to truncate the number of partial solutions stored at a node of a tree decomposition to $2^{\mathcal{O}(\mathbf{tw})}$. The dynamic programming algorithm follows the lines of proofs in [37] and is described in detail in the full version. The main technical hurdle comes from the necessity to store only the condensed counterparts of the partial solutions. The condensed graphs have only $\mathcal{O}(\mathbf{tw})$ vertices each, what is the key to obtain a linear dependency on $|V(G)|$.

5 Interval Deletion

We switch our attention to Interval Vertex Deletion and show that in this case it is unlikely to achieve any speed-up over the existing $2^{\mathcal{O}(\mathbf{tw} \log \mathbf{tw})} \cdot n$-time algorithm. We prove Theorem 1.2 via a parameterized reduction from $k \times k$ Permutation Clique, which is defined as follows.

$k \times k$ Permutation Clique
Input: Graph G over the vertex set $[k] \times [k]$.
Question: Is there a permutation $\pi\colon [k] \to [k]$ so that $(1,\pi(1)), (2,\pi(2)), \dots, (k,\pi(k))$ forms a clique in G?

Permutation gadget. We will encode a permutation $\pi\colon [k] \to [k]$ as a family of sets $N_1, N_2, \dots, N_k$ so that $N_i = \pi([i])$ (i.e., N_i is the set of i numbers appearing first in π). First, we need a gadget to verify that such a family represents some permutation.

▶ **Definition 5.1.** *For an integer k, let Y_k be a graph on a vertex set $\{y_1, y_2, \dots, y_{k+2}\}$ so that $\{y_1, y_2, \dots, y_{k+1}\}$ induces a clique and y_{k+2} is adjacent only to y_{k+1}.*

We shall enforce a linear order on $N_1, \dots, N_k$ by demanding that a particular supergraph of Y_k is interval. The corresponding interval model is depicted on Figure 2.

▶ **Lemma 5.3** (⋆). *Let $N_1, \dots, N_\ell \subseteq [k]$. Consider a graph G obtained from Y_k by inserting an independent set of vertices $x_1, \dots, x_\ell$ so that $N_G(x_i) = \{y_j \mid j \in N_i\}$. Then G is interval if and only if there exists a permutation $\pi\colon [k] \to [k]$ so that for each $i \in \ell$ it holds that $N_i = \pi([n_i])$ where $n_i = |N_i|$.*

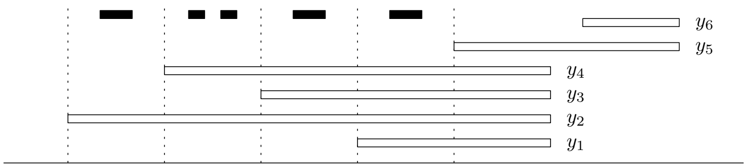

Figure 2 Illustration for Lemma 5.3. The intervals for vertices of Y_4 are blank, ordered from bottom to top. They encode permutation $(2, 4, 3, 1)$. The black intervals represent vertices x_1, x_2, x_3, x_4, x_5 with neighborhoods encoding sets $\{2\}$, $\{2, 4\}$ (twice), $\{2, 4, 3\}$, and $\{2, 4, 3, 1\}$.

Choice gadget. We need to verify that $(i, \pi(i))(j, \pi(j)) \in E(G)$ for each $1 \le i < j \le k$. As $\pi(i)$ is the only element in $N_i \setminus N_{i-1}$, the information whether $(i, \pi(i)), (j, \pi(j)) \in E(G)$ can be extracted from the tuple $(N_{i-1}, N_i, N_{j-1}, N_j)$. We construct a gadget that enforces a solution to select one such valid tuple.

We use a following convention to describe the gadgets. When P is a graph with a distinguished vertex named v and a graph H is constructed using explicit vertex-disjoint copies of the graph P, referred to as $P_1, P_2, \dots, P_\ell$, we refer to the copy of v within the subgraph P_i as $P_i[v]$. We construct the choice gadget as a path-like structure consisting of blocks, each equipped with four special vertices. These are the only vertices that later get connected to the permutation gadget. On the intuitive level, a solution should choose one block, leave its special vertices untouched, and remove the remaining special vertices. See Figure 3 for an illustration.

▶ **Definition 5.4.** *The graph P is obtained from a path $(u_1, u_2, \dots, u_9)$ by appending to u_2 two subdivided edges, one subdivided edge to u_7, and inserting edge u_4u_8.*

The choice gadget of order s is a graph constructed as follows. We begin with a vertex set $\bigcup_{i=1}^{s}\{v_i^1, v_i^2, v_i^3\} \cup \{v_{left}, v_{right}\}$. For each pair (x, y) of the form $(v_i^1, v_i^2), (v_i^2, v_i^3), (v_i^3, v_i^1), (v_i^3, v_{i+1}^1)$ as well as for (v_{left}, v_1^1), (v_s^3, v_{right}) we create two subdivided edges between x and y. We refer to the subgraph given by the two subdivided edges between x, y as $\langle x, y\rangle$. We refer to the union of $\langle v_i^1, v_i^2\rangle, \langle v_i^2, v_i^3\rangle, \langle v_i^3, v_i^1\rangle$ as Q_i.

Next, for each $i \in [s]$ we create four copies of the graph P, denoted $P_i^1, P_i^2, P_i^3, P_i^4$. We insert edges between v_i^2 and $P_i^1[u_1], P_i^2[u_1], P_i^3[u_1], P_i^4[u_1]$. We refer to vertices $P_i^\alpha[u_8]$, $P_i^\alpha[u_9]$, $\alpha \in [4]$, as respectively h_i^α, g_i^α.

The choice gadget is designed to enforce a special structure of minimum-size interval deletion sets. We exploit the fact that P contains two vertex-disjoint subgraphs with asteroidal triples (see Figure 3) so any interval deletion set in a choice gadget must contain at least two vertices from each copy of P.

We prove several properties of the choice gadget which are analogous to the properties of the gadget used by Pilipczuk in the lower bound for Planar Vertex Deletion [58]. However, in that construction every block has only one special vertex with edges leaving the gadget, while in our case there are four special vertices. We also need to ensure that when the special vertices in some block are not being removed then a solution can remove their neighbors in the gadget. (Inserting a planar graph attached to a single vertex of G does not affect planarity of G but the analogous property does not hold for the class of interval graphs.) The special structure of the graph P allows us to resolve these two issues.

▶ **Lemma 5.6** ($\star$). *Let H_s be the choice gadget of order s.*

1. *The minimal size of an interval deletion set in H_s is $10s$.*
2. *For every $i \in [s]$ there exists a minimum-size interval deletion set X in H_s such that $\{h_i^1, h_i^2, h_i^3, h_i^4\} \subseteq X$ and $\{g_j^1, g_j^2, g_j^3, g_j^4\} \subseteq X$ for each $j \neq i$.*

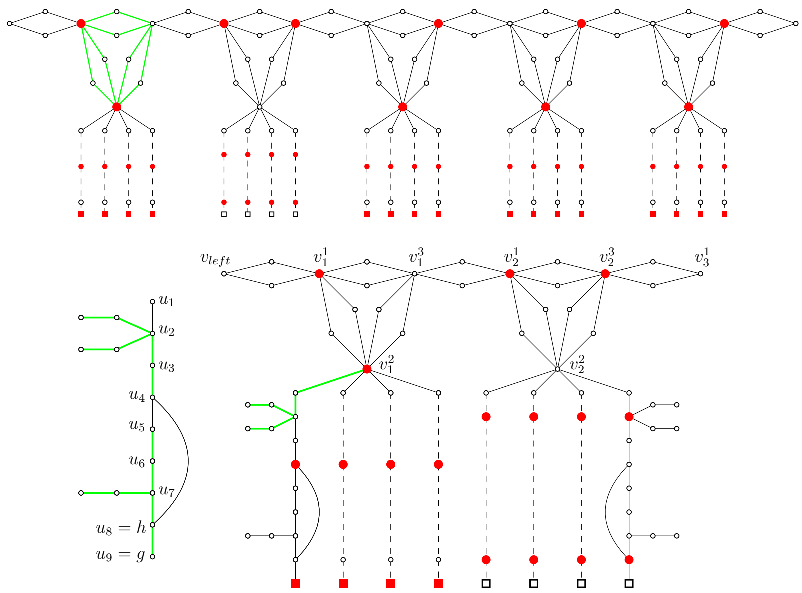

Figure 3 Top: the choice gadget H_5 with the subgraph Q_1 highlighted in green. The copies of P are sketched symbolically with dashed lines and the squares represent vertices g_i^α. The red disks and squares represent a solution constructed in Lemma 5.6(2). This solution "chooses" $i = 2$, leaves untouched the four vertices g_2^α, and removes h_2^α as well as g_i^α for $i \neq 2$. Bottom left: the graph P and vertices named h, g. Two vertex-disjoint non-interval subgraphs of P have green edges. Bottom right: a closer look at the first two blocks of H_5 with two copies of P drawn in detail. The subgraph highlighted in green witnesses that if a minimum-size solution removes g_i^α for at least one $\alpha \in [4]$ then it must also remove v_i^2, what is exploited in Lemma 5.6(3).

3. *For every minimum-size interval deletion set X in H_s there is $i \in [s]$ such that $\{g_i^1, g_i^2, g_i^3, g_i^4\} \cap X = \emptyset$.*
4. *If $s \leq 2^k$ then $\textbf{td}(H_s) \leq \textbf{td}(H_1) + k$, where $\textbf{td}(G)$ stands for the treedepth of G.*

Lokshtanov et al. [54] proved that $k \times k$ PERMUTATION CLIQUE cannot be solved in time $2^{o(k \log k)}$ assuming ETH. According to the reduction below, this also rules out running time of the form $2^{o(\textbf{td} \log \textbf{td})} \cdot n^{\mathcal{O}(1)}$ for INTERVAL VERTEX DELETION, where $\textbf{td}$ is the treedepth of the input graph. As $\textbf{tw}(G) \leq \textbf{td}(G)$, this entails the same hardness for treewidth, what proves Theorem 1.2.

▶ **Proposition 5.7.** *There is an algorithm that, given an instance (G, k) of $k \times k$ PERMUTATION CLIQUE, runs in time $2^{\mathcal{O}(k)}$ and returns an equivalent unweighted instance (H, p) of INTERVAL VERTEX DELETION such that $|V(H)| = 2^{\mathcal{O}(k)}$ and $\textbf{td}(H) = \mathcal{O}(k)$.*

Proof. For $1 \leq i < j \leq k$ and $x \neq y \in [k]$ let $\mathcal{S}_{i,x,j,y}$ be the family of tuples (S_1, S_2, S_3, S_4) of subsets of $[k]$ satisfying:

- $S_1 \subset S_2 \subseteq S_3 \subset S_4$,
- $|S_1| = i - 1$,
- $S_2 \setminus S_1 = \{x\}$,
- $|S_3| = j - 1$,
- $S_4 \setminus S_3 = \{y\}$.

Furthermore, for $1 \leq i < j \leq k$, let $\mathcal{S}_{i,j}$ be the union of $\mathcal{S}_{i,x,j,y}$ over all pairs $x \neq y \in [k]$ such that $(i, x)(j, y) \in E(G)$. Let $s_{i,j} = |\mathcal{S}_{i,j}|$ and $\rho_{i,j} \colon [s_{i,j}] \to \mathcal{S}_{i,j}$ be an arbitrary bijection. Clearly $s_{i,j} \leq 4^k k^2$.

The graph H consists of a permutation gadget Y_k and, for each $1 \le i < j \le k$, a choice gadget $C_{i,j}$ of order $s_{i,j}$. For $S \subseteq [k]$ we use shorthand $Y_k[S] = \{y_i \mid i \in S\}$. For $\ell \in [s_{i,j}]$ and $(S_1, S_2, S_3, S_4) = \rho_{i,j}(\ell)$ the vertices $C_{i,j}[g^1_\ell]$, $C_{i,j}[g^2_\ell]$, $C_{i,j}[g^3_\ell]$, $C_{i,j}[g^4_\ell]$ get connected to vertex sets $Y_k[S_1], Y_k[S_2], Y_k[S_3], Y_k[S_4]$, respectively. This finishes the construction of H. The number of vertices in H is clearly $2^{\mathcal{O}(k)}$ and the construction can be performed in time polynomial in the size of H. We set $p = 10 \cdot \sum_{1 \le i < j \le k} s_{i,j}$.

▷ Claim 5.8. If (G, k) admits a solution, then H has an interval deletion set of size p.

Proof. Let $\pi\colon [k] \to [k]$ be a permutation encoding a clique in G. By the construction, for each $1 \le i < j \le k$ we have $(\pi([i-1]), \pi([i]), \pi([j-1]), \pi([j]) \in \mathcal{S}_{i,j}$. Let $\ell \in [s_{i,j}]$ be the index mapped to this tuple by $\rho_{i,j}$. By Lemma 5.6(2) the choice gadget $C_{i,j}$ has an interval deletion set $X_{i,j} \subseteq V(C_{i,j})$ of size $10s_{i,j}$ such that $\{C_{i,j}[h^1_\ell], C_{i,j}[h^2_\ell], C_{i,j}[h^3_\ell], C_{i,j}[h^4_\ell]\} \subseteq X_{i,j}$ and $\{C_{i,j}[g^1_r], C_{i,j}[g^2_r], C_{i,j}[g^3_r], C_{i,j}[g^4_r]\} \subseteq X_{i,j}$ for each $r \ne \ell$. In other words, $X_{i,j}$ contains all vertices in $C_{i,j}$ which are adjacent to Y_k except for the $C_{i,j}$-copies of $g^1_\ell, g^2_\ell, g^3_\ell, g^4_\ell$ and $X_{i,j}$ also contains the neighbors of $C_{i,j}[g^1_\ell], C_{i,j}[g^2_\ell], C_{i,j}[g^3_\ell], C_{i,j}[g^4_\ell]$ in $C_{i,j}$.

We set $X = \bigcup_{1 \le i < j \le k} X_{i,j}$. Then the only connected component of $H - X$ which is not a connected component of any $C_{i,j} - X_{i,j}$ is given by Y_k together with an independent set of the vertices described above. The neighborhood of each such vertex in Y_k is of the form $Y_k[\pi([k'])]$ for some $0 \le k' \le k$. By Lemma 5.3 this component is an interval graph. This shows that X is indeed an interval deletion set. ◁

▷ Claim 5.9. If H has an interval deletion set of size at most p, then (G, k) admits a solution.

Proof. Let X be an interval deletion set in H. By Lemma 5.6(1) a minimum-size interval deletion set in $C_{i,j}$ has size $10s_{i,j}$. As the choice gadgets are vertex-disjoint subgraphs of H, the set X must contain exactly $10s_{i,j}$ vertices from $V(C_{i,j})$. This also implies that $V(Y_k) \cap X = \emptyset$.

Let $X_{i,j} = V(C_{i,j}) \cap X$. By Lemma 5.6(3) there exists $\ell \in [s_{i,j}]$ such that $\big\{C_{i,j}[g^1_\ell], C_{i,j}[g^2_\ell],\ C_{i,j}[g^3_\ell], C_{i,j}[g^4_\ell]\big\} \cap X_{i,j} = \emptyset$. Therefore for each pair (i,j) there is a tuple $(S^1_{i,j}, S^2_{i,j}, S^3_{i,j}, S^4_{i,j}) \in \mathcal{S}_{i,j}$ so that vertices from $C_{i,j}$ with neighborhoods $Y_k[S^1_{i,j}], Y_k[S^2_{i,j}], Y_k[S^3_{i,j}], Y_k[S^4_{i,j}]$ are present in $H - X$. By Lemma 5.3 there exists a single permutation $\pi\colon [k] \to [k]$ so that each set $S^\alpha_{i,j}$ is of the form $\pi([|S^\alpha_{i,j}|])$. By the definition of family $\mathcal{S}_{i,j}$ this implies that $(i, \pi(i))(j, \pi(j)) \in E(G)$ for each pair (i,j). Hence there is a k-clique in G. ◁

▷ Claim 5.10. The treedepth of H is $\mathcal{O}(k)$.

Proof. The treedepth of H is at most $|Y_k| = k+2$ plus $\mathbf{td}(H - Y_k)$, which equals the maximum of $\mathbf{td}(C_{i,j})$ over all employed choice gadgets $C_{i,j}$. As $s_{i,j} \le 4^k k^2$, Lemma 5.6(4) implies that $\mathbf{td}(C_{i,j}) \le 2k + 2\log_2 k + \mathcal{O}(1)$. ◁

This concludes the proof of the proposition. ◀

6 Conclusion and open problems

We have obtained ETH-tight bounds for vertex-deletion problems into the classes of chordal and interval graphs, under the treewidth parameterization. The status of the corresponding edge-deletion problems remains unclear (see [64]). The related problem, Feedback Vertex Set, can be solved using representative families within the same running time as our algorithm for ChVD [37]. However, it admits a faster deterministic algorithm based on the determinant approach [69] and an even faster randomized algorithm based on the Cut & Count technique [35]. Could ChVD also be amenable to one of those techniques?

Our algorithm for ChVD is based on a novel connection between chordal graphs and graphic matroids, which might come in useful in other settings. In particular, we ask whether this insight can be leveraged to improve the running time for ChVD parameterized by the solution size k, where the current-best algorithm runs in time $2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)}$ [29]. A direct avenue for a potential improvement would be to reduce the problem in time $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ to the case with treewidth $\mathcal{O}(k)$ and then apply Theorem 1.1. Such a strategy has been employed in the state-of-the-art algorithm for Planar Vertex Deletion parameterized by the solution size [47].

References

1. Akanksha Agrawal, Sudeshna Kolay, Daniel Lokshtanov, and Saket Saurabh. A faster FPT algorithm and a smaller kernel for block graph vertex deletion. In Evangelos Kranakis, Gonzalo Navarro, and Edgar Chávez, editors, *LATIN 2016: Theoretical Informatics – 12th Latin American Symposium, Ensenada, Mexico, April 11-15, 2016, Proceedings*, volume 9644 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2016. doi:10.1007/978-3-662-49529-2_1.
2. Akanksha Agrawal, Daniel Lokshtanov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Erdös-Pósa property of obstructions to interval graphs. In Rolf Niedermeier and Brigitte Vallée, editors, *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*, volume 96 of *LIPIcs*, pages 7:1–7:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.STACS.2018.7.
3. Akanksha Agrawal, Daniel Lokshtanov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Feedback vertex set inspired kernel for chordal vertex deletion. *ACM Trans. Algorithms*, 15(1):11:1–11:28, 2019. doi:10.1145/3284356.
4. Akanksha Agrawal, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Interval vertex deletion admits a polynomial kernel. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, pages 1711–1730, USA, 2019. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611975482.103.
5. Jungho Ahn, Eduard Eiben, O joung Kwon, and Sang il Oum. A Polynomial Kernel for 3-Leaf Power Deletion. In Javier Esparza and Daniel Kráľ, editors, *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2020.5.
6. Jungho Ahn, Eun Jung Kim, and Euiwoong Lee. Towards constant-factor approximation for chordal/distance-hereditary vertex deletion. *Algorithmica*, 84(7):2106–2133, 2022. doi:10.1007/s00453-022-00963-7.
7. Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '21, pages 522–539, USA, 2021. Society for Industrial and Applied Mathematics. doi:10.5555/3458064.3458096.
8. Amotz Bar-Noy, Reuven Bar-Yehuda, Ari Freund, Joseph (Seffi) Naor, and Baruch Schieber. A unified approach to approximating resource allocation and scheduling. *J. ACM*, 48(5):1069–1090, September 2001. doi:10.1145/502102.502107.
9. Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. A complexity dichotomy for hitting connected minors on bounded treewidth graphs: the chair and the banner draw the boundary. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 951–970. SIAM, 2020. doi:10.1137/1.9781611975994.57.
10. Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. I. General upper bounds. *SIAM J. Discret. Math.*, 34(3):1623–1648, 2020. doi:10.1137/19M1287146.

11 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. II. Single-exponential algorithms. *Theor. Comput. Sci.*, 814:135–152, 2020. `doi:10.1016/j.tcs.2020.01.026`.

12 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. III. Lower bounds. *J. Comput. Syst. Sci.*, 109:56–77, 2020. `doi:10.1016/j.jcss.2019.11.002`.

13 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall PTR, USA, 1st edition, 1998. URL: `https://dl.acm.org/doi/10.5555/551884`.

14 Seymour Benzer. On the topology of the genetic fine structure. *Proceedings of the National Academy of Sciences of the United States of America*, 45(11):1607–1620, 1959. URL: `http://www.jstor.org/stable/90127`.

15 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)*, volume 180 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:17, 2020. `doi:10.4230/LIPIcs.IPEC.2020.3`.

16 Jean R. S. Blair and Barry Peyton. An introduction to chordal graphs and clique trees. In Alan George, John R. Gilbert, and Joseph W. H. Liu, editors, *Graph Theory and Sparse Matrix Computation*, pages 1–29, New York, NY, 1993. Springer New York.

17 Ivan Bliznets, Marek Cygan, Pawel Komosa, Michal Pilipczuk, and Lukás Mach. Lower bounds for the parameterized complexity of minimum fill-in and other completion problems. *ACM Trans. Algorithms*, 16(2):25:1–25:31, 2020. `doi:10.1145/3381426`.

18 Ivan Bliznets, Fedor V Fomin, Michał Pilipczuk, and Yngve Villanger. Largest chordal and interval subgraphs faster than 2^n. *Algorithmica*, 76(2):569–594, 2016.

19 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015. `doi:10.1016/j.ic.2014.12.008`.

20 Hans L. Bodlaender, Pål Gronås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michał Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016. `doi:10.1137/130947374`.

21 Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (Meta) Kernelization. *J. ACM*, 63(5):44:1–44:69, 2016. `doi:10.1145/2973749`.

22 Marthe Bonamy, Lukasz Kowalik, Jesper Nederlof, Michal Pilipczuk, Arkadiusz Socala, and Marcin Wrochna. On directed feedback vertex set parameterized by treewidth. In Andreas Brandstädt, Ekkehard Köhler, and Klaus Meer, editors, *Graph-Theoretic Concepts in Computer Science – 44th International Workshop, WG 2018, Cottbus, Germany, June 27-29, 2018, Proceedings*, volume 11159 of *Lecture Notes in Computer Science*, pages 65–78. Springer, 2018. `doi:10.1007/978-3-030-00256-5_6`.

23 Édouard Bonnet, Nick Brettell, O joung Kwon, and Dániel Marx. Generalized Feedback Vertex Set Problems on Bounded-Treewidth Graphs: Chordality Is the Key to Single-Exponential Parameterized Algorithms. In Daniel Lokshtanov and Naomi Nishimura, editors, *12th International Symposium on Parameterized and Exact Computation (IPEC 2017)*, volume 89 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 7:1–7:13, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.IPEC.2017.7`.

24 Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph classes: a survey*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999. `doi:10.1137/1.9780898719796`.

25 Peter Buneman. A characterisation of rigid circuit graphs. *Discrete Math.*, 9(3):205–212, September 1974. `doi:10.1016/0012-365X(74)90002-8`.

26 Leizhen Cai. Parameterized complexity of vertex colouring. *Discrete Applied Mathematics*, 127(3):415–429, 2003. doi:10.1016/S0166-218X(02)00242-1.

27 Yixin Cao. Linear recognition of almost interval graphs. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1096–1115. SIAM, 2016. doi:10.1137/1.9781611974331.ch77.

28 Yixin Cao and Dániel Marx. Interval deletion is fixed-parameter tractable. *ACM Trans. Algorithms*, 11(3), January 2015. doi:10.1145/2629595.

29 Yixin Cao and Dániel Marx. Chordal editing is fixed-parameter tractable. *Algorithmica*, 75(1):118–137, May 2016. doi:10.1007/s00453-015-0014-x.

30 Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic – A Language-Theoretic Approach*, volume 138 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 2012. doi:10.1017/CBO9780511977619.

31 Radu Curticapea, Nathan Lindzey, and Jesper Nederlof. A tight lower bound for counting hamiltonian cycles via matrix rank. In *Proceedings of the 2018 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1080–1099, 2018. doi:10.1137/1.9781611975031.70.

32 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

33 Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Fast hamiltonicity checking via bases of perfect matchings. *J. ACM*, 65(3):12:1–12:46, 2018. doi:10.1145/3148227.

34 Marek Cygan, Dániel Marx, Marcin Pilipczuk, and Michał Pilipczuk. Hitting forbidden subgraphs in graphs of bounded treewidth. *Information and Computation*, 256:62–82, 2017. doi:10.1016/j.ic.2017.04.009.

35 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michał Pilipczuk, Johan M. M. Van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. *ACM Trans. Algorithms*, 18(2), March 2022. doi:10.1145/3506707.

36 Erik D. Demaine and MohammadTaghi Hajiaghayi. The bidimensionality theory and its algorithmic applications. *The Computer Journal*, 51(3):292–302, 2008. doi:10.1093/comjnl/bxm033.

37 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Representative sets of product families. In Andreas S. Schulz and Dorothea Wagner, editors, *Algorithms – ESA 2014*, pages 443–454, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. doi:10.1007/978-3-662-44777-2_37.

38 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016. doi:10.1145/2886094.

39 Fedor V. Fomin and Yngve Villanger. Subexponential parameterized algorithm for minimum fill-in. *SIAM Journal on Computing*, 42(6):2197–2216, 2013. doi:10.1137/11085390X.

40 Fănică Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B*, 16(1):47–56, 1974. doi:10.1016/0095-8956(74)90094-X.

41 Pinar Heggernes, Pim van 't Hof, Bart M.P. Jansen, Stefan Kratsch, and Yngve Villanger. Parameterized complexity of vertex deletion into perfect graph classes. *Theoretical Computer Science*, 511:172–180, 2013. Exact and Parameterized Computation. doi:10.1016/j.tcs.2012.03.013.

42 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

43 Ashwin Jacob, Fahad Panolan, Venkatesh Raman, and Vibha Sahlot. Structural parameterizations with modulator oblivion. 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 19:1–19:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.IPEC.2020.19.

44 Hugo Jacob, Thomas Bellitto, Oscar Defrain, and Marcin Pilipczuk. Close Relatives (Of Feedback Vertex Set), Revisited. In Petr A. Golovach and Meirav Zehavi, editors, *16th International Symposium on Parameterized and Exact Computation (IPEC 2021)*, volume 214 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:15, Dagstuhl, Germany, 2021. `doi:10.4230/LIPIcs.IPEC.2021.21`.

45 Bart M. P. Jansen, Jari J. H. de Kroon, and Michał Włodarczyk. Vertex deletion parameterized by elimination distance and even less. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 1757–1769, New York, NY, USA, 2021. Association for Computing Machinery. `doi:10.1145/3406325.3451068`.

46 Bart M. P. Jansen, Jari J. H. de Kroon, and Michal Wlodarczyk. Vertex deletion parameterized by elimination distance and even less. *CoRR*, abs/2103.09715, 2021. `arXiv:2103.09715v4`.

47 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`.

48 Bart M. P. Jansen and Marcin Pilipczuk. Approximation and kernelization for chordal vertex deletion. *SIAM J. Discret. Math.*, 32(3):2258–2301, 2018. `doi:10.1137/17M112035X`.

49 J.Mark Keil. Finding hamiltonian circuits in interval graphs. *Information Processing Letters*, 20(4):201–206, 1985. `doi:10.1016/0020-0190(85)90050-X`.

50 David Kendall. Incidence matrices, interval graphs and seriation in archeology. *Pacific Journal of mathematics*, 28(3):565–570, 1969.

51 Eun Jung Kim and O-joung Kwon. Erdős-Pósa property of chordless cycles and its applications. *J. Comb. Theory, Ser. B*, 145:65–112, 2020. `doi:10.1016/j.jctb.2020.05.002`.

52 Lefteris M. Kirousis and Christos H. Papadimitriou. Interval graphs and searching. *Discrete Mathematics*, 55(2):181–184, 1985. `doi:10.1016/0012-365X(85)90046-9`.

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

54 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Slightly superexponential parameterized problems. *SIAM J. Comput.*, 47(3):675–702, 2018. `doi:10.1137/16M1104834`.

55 Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. On the hardness of eliminating small induced subgraphs by contracting edges. In Gregory Z. Gutin and Stefan Szeider, editors, *Parameterized and Exact Computation – 8th International Symposium, IPEC 2013, Sophia Antipolis, France, September 4-6, 2013, Revised Selected Papers*, volume 8246 of *Lecture Notes in Computer Science*, pages 243–254. Springer, 2013. `doi:10.1007/978-3-319-03898-8_21`.

56 Dániel Marx. Four shorts stories on surprising algorithmic uses of treewidth. In Fedor V. Fomin, Stefan Kratsch, and Erik Jan van Leeuwen, editors, *Treewidth, Kernels, and Algorithms – Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, volume 12160 of *Lecture Notes in Computer Science*, pages 129–144. Springer, 2020. `doi:10.1007/978-3-030-42071-0_10`.

57 Dániel Marx, Barry O'Sullivan, and Igor Razgon. Finding small separators in linear time via treewidth reduction. *ACM Trans. Algorithms*, 9(4):30:1–30:35, 2013. `doi:10.1145/2500119`.

58 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`.

59 Michał Pilipczuk. Problems parameterized by treewidth tractable in single exponential time: A logical approach. In Filip Murlak and Piotr Sankowski, editors, *Mathematical Foundations of Computer Science 2011*, pages 520–531, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. `doi:10.1007/978-3-642-22993-0_47`.

60 R. C. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957. `doi:10.1002/j.1538-7305.1957.tb01515.x`.

61 N. Robertson and P.D. Seymour. Graph minors XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995. `doi:10.1006/jctb.1995.1006`.

62 Neil Robertson and P.D Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986. doi:10.1016/0196-6774(86)90023-4.

63 Donald J Rose. A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. In *Graph theory and computing*, pages 183–217. Elsevier, 1972.

64 Toshiki Saitoh, Ryo Yoshinaka, and Hans L. Bodlaender. Fixed-treewidth-efficient algorithms for edge-deletion to interval graph classes. In Ryuhei Uehara, Seok-Hee Hong, and Subhas C. Nandy, editors, *WALCOM: Algorithms and Computation – 15th International Conference and Workshops, 2021, Yangon, Myanmar*, volume 12635 of *Lecture Notes in Computer Science*, pages 142–153. Springer, 2021. doi:10.1007/978-3-030-68211-8_12.

65 Ignasi Sau and Uéverton dos Santos Souza. Hitting Forbidden Induced Subgraphs on Bounded Treewidth Graphs. In Javier Esparza and Daniel Kráľ, editors, *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 82:1–82:15, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2020.82.

66 Hadas Shachnai and Meirav Zehavi. Representative families: A unified tradeoff-based approach. *Journal of Computer and System Sciences*, 82(3):488–502, 2016. doi:10.1016/j.jcss.2015.11.008.

67 Johan MM Van Rooij, Hans L Bodlaender, and Peter Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In *European Symposium on Algorithms*, pages 566–577. Springer, 2009. doi:10.1007/978-3-642-04128-0_51.

68 Pim van 't Hof and Yngve Villanger. Proper interval vertex deletion. *Algorithmica*, 65(4):845–867, 2013. doi:10.1007/s00453-012-9661-3.

69 Michał Włodarczyk. Clifford algebras meet tree decompositions. *Algorithmica*, 81(2):497–518, 2019. doi:10.1007/s00453-018-0489-3.

70 Mihalis Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM Journal on Algebraic Discrete Methods*, 2(1):77–79, 1981. doi:10.1137/0602010.

71 Meirav Zehavi. Mixing color coding-related techniques. In *Algorithms-ESA 2015*, pages 1037–1049. Springer, 2015. doi:10.1007/978-3-662-48350-3_86.