Chordless Cycle Packing Is Fixed-Parameter Tractable

Dániel Marx
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
marx@cispa.saarland

Abstract

A *chordless cycle* or *hole* in a graph G is an induced cycle of length at least 4. In the Hole Packing problem, a graph G and an integer k is given, and the task is to find (if exists) a set of k pairwise vertex-disjoint chordless cycles. Our main result is showing that Hole Packing is fixed-parameter tractable (FPT), that is, can be solved in time $f(k)n^{O(1)}$ for some function f depending only on k.

2012 ACM Subject Classification Theory of computation → Data structures design and analysis; Theory of computation → Graph algorithms analysis

Keywords and phrases chordal graphs, packing, fixed-parameter tractability

Digital Object Identifier 10.4230/LIPIcs.ESA.2020.71

Funding Supported by the European Research Council (ERC) consolidator grant No. 725978 SYSTEMATICGRAPH.

Acknowledgements The author is very grateful to O-joung Kwon and Eun Jung Kim for helpful comments on the manuscript.

1 Introduction

The area of graph modification problems contains algorithmic tasks of the following form: given a graph G, find the minimum number of allowed editing operations to make the graph belong to a certain target graph class $\mathcal{G}$. For example, if we allow only vertex deletions and the target graph class is the set of edgeless graphs, forests, directed acyclic graphs, bipartite graphs, then we get well-known Vertex Cover, Feedback Vertex Set, Directed Feedback Vertex Set, Bipartite Deletion problems, respectively. Most of the natural graph modification problems are NP-hard [21, 30, 31]. However, there is a large literature on the fixed-parameter tractability of graph modification problems (see, e.g., [4, 8, 12, 13, 15]). Several problems of this form are known to be solvable in time $f(k)n^{O(1)}$, where k is the number of editing operations allowed (e.g., maximum number of vertices to be deleted) and f is a computable function depending only on k [9].

Let us consider the $\mathcal{G}$ Vertex Deletion problem where, given a graph G and an integer k, the task is to delete k vertices to make the graph belong to class $\mathcal{G}$. If $\mathcal{G}$ is closed under taking induced subgraphs, then there is a (finite or infinite) set $\mathcal{F}$ of obstructions such that a graph is in $\mathcal{G}$ if and only if it does not have an induced subgraph isomorphic to a member of $\mathcal{F}$. Then $\mathcal{G}$ Vertex Deletion can be equivalently expressed as finding k vertices that cover every induced copy of a member of $\mathcal{F}$. For many natural graph properties, the obstruction set $\mathcal{F}$ contains graphs that are simple to describe. For example, the problems Vertex Cover, Feedback Vertex Set, Directed Feedback Vertex Set, and Bipartite Deletion correspond to covering every edge, undirected cycle, directed cycle, and odd cycle, respectively.

Given the interpretation of $\mathcal{G}$ Vertex Deletion as covering objects from the obstruction set $\mathcal{F}$, there is natural dual problem: in the $\mathcal{F}$ Packing problem a graph G and an integer k are given, the task is to find k vertex-disjoint induced subgraphs isomorphic to members of $\mathcal{F}$.

In many cases, packing problems seem to be harder than the corresponding covering problems. First of all, if the graph class $\mathcal{G}$ is recognizable in polynomial-time, then the covering problem can be solved in $n^{O(k)}$ time by brute force, while there is no such immediate argument for the packing problem, even when the class $\mathcal{F}$ consists of very simple objects, such as cycles. For example, Directed Feedback Vertex Set is FPT [6], while the dual problem Directed Cycle Packing is W[1]-hard [28] and it requires a highly nontrivial result to show that it is polynomial-time solvable for fixed k [25]. Even when both problems are FPT, the techniques behind the algorithms could be significantly different. Bipartite Deletion has a very elegant elementary FPT algorithm using iterative compression [26], while the fixed-parameter tractability of the dual problem Odd Cycle Packing required the use of sophisticated techniques, including the introduction of odd minors [18, 19]. Another aspect from which the packing problem proved to be more difficult is the existence of polynomial kernels. For example, Feedback Vertex set (that is, covering cycles) admits a polynomial kernel [29], while the dual problem of finding k vertex-disjoint cycles does not have a polynomial kernel, under the standard complexity assumption $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$ [3].

There is a natural combinatorial question connecting the covering and packing problems. A classic result of Erdős and Pósa [11] shows that if the maximum number of vertex-disjoint cycles in graph G is k, then every cycle of G can be covered by $O(k \log k)$ vertices. A similar question can be asked about other obstructions, connecting the packing and covering problems: if the maximum number of disjoint obstructions from the set $\mathcal{F}$ is at most k, then is it true that every obstruction can be covered by $f(k)$ vertices for some function f? A positive answer is known for example when $\mathcal{F}$ is the set of edges (easy), undirected cycles going through a set S [17, 23], or directed cycles [25], but it is known that no such Erdős-Pósa property holds for odd cycles [24]. Let us observe that, as the cases of odd cycles and directed cycles show, even when the covering problem is FPT, the existence of the Erdős-Pósa property does not give a good prediction on the fixed-parameter tractability of the packing problem.

Chordal graphs. After this general introduction, let us turn our attention to chordal graphs, the main topic of the current paper. A *chordless cycle* or *hole* in a graph G is an induced cycle of length at least 4 (for brevity, we will use the term "hole" throughout the paper). A graph is *chordal* if it does not contain any hole. Chordal graphs form a well-known class of perfect graphs and it is known that a graph is chordal if and only if it can be represented as the intersection graph of a set of subtrees of a tree [14]. Chordal graphs can be recognized in linear time [27]. In the Chordal Vertex Deletion problem, a graph G and an integer k are given, and the task is to find a set S of at most k vertices such that $G - S$ is chordal. While the problem can be solved in time $n^{O(k)}$ by trying every subset of size at most k and then testing for chordality, it is also known to be FPT.

▶ **Theorem 1** ([1, 5, 16, 22]). Chordal Vertex Deletion *is FPT.*

Kim and Kwon gave a constructive proof showing that holes have the Erdős-Pósa property.

▶ **Theorem 2** (Kim and Kwon [20]). *There is a polynomial-time algorithm that, given a graph G and integer k, produces either*
1. *a set of $k+1$ disjoint holes, or*
2. *a set of $O(k^2 \log k)$ vertices covering every hole.*

Our main result concerns the Hole Packing problem, where given a graph G and an integer k, the task is to find a set of k pairwise vertex-disjoint holes.

▶ **Theorem 3** (Main Result). Hole Packing *is FPT.*

Let us remark that it is known that Chordal Vertex Deletion admits a polynomial kernel [1,16], while an easy reduction gives negative evidence for Hole Packing. Bodlaender et al. [3] showed that the problem of finding k pairwise vertex-disjoint cycles does not admit a polynomial kernel under the complexity assumption $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$. If we subdivide every edge of a simple graph, then every cycle has length at least 6, which means that the holes of the new graph are in one-to-one correspondence with the cycles of the original graph. Therefore, the result of Bodlaender et al. [3] immediately implies that Hole Packing has no polynomial kernel, assuming $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$. This can be seen as an indication that the dual problem Hole Packing is more challenging than Chordal Vertex Deletion, and it can be expected that more involved algorithmic ideas are needed for this problem.

Our techniques. To explain the main ideas and challenges behind the algorithm of Theorem 3, let us briefly overview the Chordal Vertex Deletion algorithm of Marx [22]; our algorithm mirrors the technical ideas from that result up to a certain point, but then it needs to deviate from it significantly. When solving Chordal Vertex Deletion, the standard technique of iterative compression [26] allows us to assume that we know a set W of $k+1$ vertices such that $G - W$ is chordal and furthermore we can assume that the solution S of size k we are looking for is disjoint from W. If the size of the largest clique in $G - W$ can be bounded by a function of k, then the treewidth of the chordal graph $G - W$ and hence also the treewidth of the slightly larger (not necessarily chordal) G can be bounded by a function of k. In this case, the problem can be solved on the graph G using standard algorithmic techniques on graphs of bounded treewidth, for example, using Courcelle's Theorem [7].

If $G - W$ has a large clique K, then, intuitively, we want to argue that a large part of the clique is not really important for the problem. More formally, we want to identify a vertex $v \in K$ such that removing k from G does not make the problem any easier. We say that v is *irrelevant* if for every set of S of size at most k disjoint from W, if there is a hole in $G - S$, then there is a hole in $G - (S \cup \{v\})$ as well. The algorithm of Marx [22] marks a certain number of vertices in K as important and then it is argued that every other vertex $v \in K$ is irrelevant in this sense. The proof is mostly a rerouting argument: if there is a hole going through v in $G - S$, then it has to be shown that the hole can be modified to avoid v.

To solve the Hole Packing problem, let us observe that Theorem 2 allows us to assume that we have a set W of $O(k^2 \log k)$ vertices such that $G - W$ is chordal: if the algorithm of Theorem 2 terminates with Outcome 1, then we are done. If $G - W$ has maximum clique size bounded by function of k, then G has bounded treewidth and we can use standard techniques to find a set of k vertex-disjoint holes. Thus our goal again is to argue that we can find an irrelevant vertex v in a large clique K. But now our notion of irrelevant vertex is different: for Chordal Vertex Deletion, a vertex needed to be irrelevant with respect to a deletion set S of at most k vertices, while for Hole Packing, vertex v needs to be irrelevant with respect to a set of $k-1$ holes. Formally, now we can say that v is *irrelevant* if whenever G has a set $\mathcal{H}$ of k disjoint holes, then there is such a set avoiding v. The set W splits $\mathcal{H}$ into at most k induced paths. Therefore, we again need a rerouting argument: the induced path going through v needs to be rerouted to avoid the other at most $|W| - 1$ induced paths.

Rerouting a path to avoid a bounded number of induced paths seems to be a significantly more challenging task compared to avoiding a bounded number of vertices: the paths can be arbitrarily long and we cannot bound the number of vertices they contain. However, it is useful to observe that an induced path can contain at most two vertices from a clique. Therefore, looking locally at a clique, avoiding a bounded number of induced paths is not all that different from avoiding a bounded number of vertices. Indeed, it seems that we can reuse many of the technical ideas from [22] to Hole Packing (we found it convenient to use somewhat different notation and streamlined some of the proofs, but we face essentially the same difficulties and similar arguments are needed). However, we did not manage to fully translate the approach to Hole Packing and to reduce the maximum clique size of $G - W$ (and hence the treewidth of G) to be bounded by a function of k. There is a particular situation where no vertex of a large clique can be declared irrelevant under our definition; Section 5 describes an example.

The first part of our algorithm uses these irrelevant vertex arguments to find a bounded-treewidth subgraph of G that contains the solution, except a few vertices of the solution that appear in a very specific situation (Section 3). This part of the proof uses ideas similar to the Chordal Vertex Deletion algorithm of [22], with appropriate modifications to account for induced paths. The main technical novelty of the paper appears in the way the problem is treated after this step. We find a way of encoding the problem in a bounded-treewidth labeled graph; however, for this to work, we need to leave the setting of the Hole Packing problem and introduce a technical variant of the problem which we call Special Hole Packing (Section 4). This problem involves finding k pairwise disjoint holes subject to certain technical conditions on the labels of vertices used by the holes. We show that when we move to this problem, then the large cliques can be reduced. Essentially, if there is a vertex v in a large clique K, then we look at the subtree intersection representation of the chordal graph $G - W$, and replace the subtree T_v representing v with a set of vertices representing the leaves of T_v. Applying this operation to every vertex of every large clique results in a graph with bounded treewidth. An appropriate choice of labeling, provided by the Color Coding [2] method, ensures that the reduction results in an instance of Special Hole Packing whose solution gives a solution to the original problem.

2 Preliminaries

We use standard graph-theoretic notation, see e.g. [10]. For background on parameterized algorithms, see [9]. In this section, we only discuss notation and basic results related to treewidth and chordal graphs.

Treewidth. A *tree decomposition* of a graph G is a pair $(T, \mathcal{B})$ in which T is a tree and $\mathcal{B} = \{B_t \mid t \in V(T)\}$ is a family of subsets of $V(G)$ such that

1. $\bigcup_{t\in V(T)} B_i = V(G)$;
2. for each edge $e = uv \in E(G)$, there exists an $t \in V(T)$ such that both u and v belong to B_t; and
3. the set of nodes $\{t \in V(T) \mid v \in B_t\}$ forms a connected subtree of T for every $v \in V(G)$.

To distinguish between vertices of the original graph G and vertices of T in the tree decomposition, we call vertices of T *nodes* and their corresponding B_i's *bags*. The *width* of the tree decomposition is the maximum size of a bag in $\mathcal{B}$ minus 1. The *treewidth* of a graph G, denoted by $\text{tw}(G)$, is the minimum width over all possible tree decompositions of G.

Sentences in *Monadic Second Order Logic of Graphs* (MSO) contain quantifiers, logical connectives ($\neg$, $\vee$, and $\wedge$), vertex variables, vertex set variables, binary relations $\in$ and $=$, and the atomic formula $E(u, v)$ expressing that u and v are adjacent. Courcelle's Theorem states that if a graph property can be described in this language, then this description can be turned into an algorithm:

▶ **Theorem 4** (Courcelle [7]). *If a graph property can be described as a formula ϕ in the Monadic Second Order Logic of Graphs, then it can be recognized in time $f(|\phi|, \mathrm{tw}(G)) \cdot (|E(G)| + |V(G)|)$ if a given graph G has this property.*

Courcelle's Theorem works also in the more general setting of relational structures. For our purposes, it will be sufficient to know that the result can be extended in such a way that the input graph G comes with a labeling $\lambda : V(G) \to [c]$ of the vertices and the formula ϕ may contain unary predicates $C_1(v)$, ..., $C_c(v)$, stating that vertex v has a label.

Chordal graphs. A *chordless cycle* or *hole* in a graph G is an induced cycle of length at least 4. A graph is *chordal* if it does not contain any hole. It is well known that a graph is chordal if and only if it can be represented as the intersection graph of subtrees of a tree. That is, every chordal graph G can be represented by a tree T and a subtree T_v of T corresponding to every $v \in V(G)$ such that $u, v \in V(G)$ are adjacent if and only if T_u and T_v share at least one node in T. Equivalently, a graph G is chordal if and only if it has a tree decomposition $(T, \mathcal{B})$ where every bag B_t induces a clique $G[B_t]$ for every $t \in V(T)$. Such a tree decomposition is also called a *clique tree decomposition*. We will use the well-known fact that if K is a clique in the chordal graph, then the clique tree decomposition contains a node t with $K \subseteq B_t$.

The following lemma is straightforward:

▶ **Lemma 5.** *Let x and y be two nonadjacent neighbors of v and let P be an $x - y$ path whose internal vertices are not in the closed neighborhood of v. Then the graph induced by $V(P) \cup \{v\}$ contains a hole.*

Induced paths. Suppose that $\mathcal{H}$ is a collection of holes in a graph G and W is a set of vertices that intersects each hole in $\mathcal{H}$. Then W splits each hole in $\mathcal{H}$ into some number of induced paths, that is, what remains from $\mathcal{H}$ is a collection of at most $|W|$ induced paths. This motivates the following definition.

▶ **Definition 6.** *A set X of vertices of a graph G is a k-*IP set *if it has partition $(X_1, \ldots, X_{k'})$ into $k' \leq k$ classes such that each $G[X_i]$ is an induced path (possibly of length 0, i.e., consisting only of a single vertex).*

Note that this definition allows the existence of arbitrary edges between X_i and X_j for $i \neq j$. The first basic observation is that such a set has small intersection with a clique.

▶ **Lemma 7.** *Let X be a k-IP set in a graph G and let K be a clique in G. Then X contains at most $2k$ vertices of K.*

Proof. Let $(X_1, \ldots, X_{k'})$ be a partition of X into induced paths. It is clear that an induced path can contain at most two vertices of the clique K, hence $|X \cap K| \leq 2k' \leq 2k$. ◀

The second observation is that a k-IP set can enter only a bounded number of components after the removal of a clique (or, more generally, if the neighborhood of each resulting component is a clique).

▶ **Lemma 8.** *Let X be a k-IP set in graph G and let Y be a set of vertices such that it is true for every component C of $G - Y$ that the neighborhood of C in Y is a clique. Then X intersects at most $2k$ components of $G - Y$.*

Proof. Let $(X_1, \dots, X_{k'})$ be a partition of X into induced paths. We claim that each X_i can intersect at most two components of $G - Y$. If X_i intersects three components, then it is true for some component C that the induced path $G[P_i]$ enters C from Y and then later leaves C to Y. But the neighborhood of C in Y is a clique, contradicting the assumption that $G[X_i]$ is an induced path. Thus in total X can intersect at most $2k' \leq 2k$ components of $G - Y$. ◀

3 Part 1: Treewidth reduction (almost)

Given an instance (G, k) of Hole Packing, an application of Theorem 2 gives us a set W of $O(k^2 \log k)$ vertices such that $G - W$ is chordal. The main result of this section is a marking procedure (Lemma 9) that identifies a bounded-treewidth part of the graph that *almost* contains the solution. As $G - W$ is chordal, every hole in G contains at least one vertex of W. A *special hole* is a hole that contains exactly one vertex of W. The *special vertices* of a special hole H going through $w \in W$ are the two neighbors of w in H. Note that, as every hole has length at least 4, the two special vertices of a special hole are not adjacent.

▶ **Lemma 9.** *Let G be graph and W be a set of vertices such that $G - W$ is chordal, and let k be an integer. In polynomial time, we can find a set $S \subseteq V(G) \setminus W$ of vertices such that the following holds:*
1. *If G has a set of k pairwise disjoint holes, then there is such a set $\mathcal{H}$ where every vertex of every hole is in $S \cup W$, except possibly some of the special vertices of the special holes.*
2. *The maximum clique size in $G[S]$ is $12(|W|+2)^4$.*

The proof of Lemma 9 starts with $S = V(G) \setminus W$ and if $G[S]$ contains a large clique K, then it tries to identify a vertex $v \in K$ that can be excluded from S without violating Requirement 1. Towards this goal, the following lemma either finds a collection of paths that are useful for creating holes going through K (Outcome 1), or marks a bounded-sized set M of vertices that are somehow important in the clique and a path reaching the clique at a vertex of $K \setminus M$ can be rerouted to reach the clique at some other vertex (Outcome 2).

▶ **Lemma 10.** *Let K be a clique in a chordal graph G, A be a set of vertices, and k be an integer. There is a polynomial-time algorithm that produces one of the following two outcomes:*
1. *A collections $\mathcal{P}$ of paths such that*
   - *every path in $\mathcal{P}$ is a path of length at least one from A to K with exactly one vertex in A,*
   - *the first endpoints of the paths in $\mathcal{P}$ form an independent set of A, and*
   - *if X is a k-IP set, then at least* two *of the paths in $\mathcal{P}$ are disjoint from X.*
2. *A subset M of K having size at most $(2k+1)(4k+2)$ such that following holds: if P is a path of length at least one from $a \in A$ to $v \in K \setminus M$ having exactly one vertex in A, and X is a k-IP set disjoint from $V(P)$, then there is a path P' from vertex $a \in A$ to a vertex of K such that P' is disjoint from $X \cup \{v\}$ and moreover $V(P') \setminus V(P) \subseteq K$. (Note that path P' can have length 0 and may contain more than one vertex from A.)*

Proof. Let us consider a subtree representation of the chordal graph G over the tree T. For every node $x \in V(T)$, let us denote by bag B_x the set of those vertices whose subtrees contain x. Let U be the set of nodes y for which $|B_y \cap K| > 2k+1$. It is easy to see that U induces a connected subtree of T. Let subtrees $T_1, \dots, T_c$ be the components of $T - U$ and let C_i contain those vertices of G whose subtrees are completely contained in the subtree T_i. Every T_i has a unique node w_i that is adjacent to U. Let us observe that the neighborhood of every C_i is a clique: if a vertex v has a neighbor in C_i, but it is not itself in C_i, then the subtree T_v has to contain a node of T_i and a node not in T_i. This is only possible if the subtree T_v contains node w_i and such vertices v form a clique.

For every $1 \le i \le c$, let $M_i \subseteq K$ be defined the following way. If for a vertex $v \in K$, there is a path P^i_v of length at least one from a vertex of A to v such that P^i_v has exactly one vertex in A and every vertex of P^i_v except v is in C_i, then we put v into M_i. For the rest of the proof, let us fix such a path P^i_v for every vertex $v \in M_i$. Let us observe that if $v \in M_i$, then the subtree of v contains a node of T_i and a node of U, hence it contains the node w_i. As $w_i \notin U$ by definition, there are at most $2k+1$ vertices of K whose subtree contains w_i and $|M_i| \le 2k+1$ follows.

Let $M = \bigcup_{i=1}^{c} M_i$. We consider two cases. If $|M| > (2k+1)(4k+1)$, then we claim the set of paths required by Outcome 1 exist. In this case, a simple greedy selection argument shows the existence of a subset of $t = 4k+2$ paths $P^{i_1}_{v_1}$, $\dots$, $P^{i_t}_{v_t}$ of the paths defined above such that the integers i_1, $\dots$, i_t and the vertices v_1, $\dots$, v_t are all distinct. That is, if we have already selected $j \le 4k+1$ of these paths, then they together can block at most $(2k+1)j \le (2k+1)(4k+1)$ vertices of M, hence we can add one more path $P^{i_{j+1}}_{v_{j+1}}$ to our collection. We claim that these paths satisfy the requirements. Path $P^{i_j}_{v_j}$ starts in a vertex of $A \cap C_{i_j}$, whose subtree is fully contained in T_{i_j}. As i_1, $\dots$, i_t are distinct integers, the start vertices of these paths are independent vertices of A, as required. Let now X be a k-IP. By Lemma 7, X contains at most $2k$ vertices of K, thus it can intersect at most $2k$ of the vertices v_1, $\dots$, v_h. By Lemma 8, X can intersect at most $2k$ of the sets C_{i_1}, $\dots$, C_{i_h}: recall that the neighborhood of each C_j is a clique. In summary, X contain at most $2k$ of v_{i_1}, $\dots$, v_{i_t} and intersects at most $2k$ of C_{i_t}, $\dots$, C_{i_t}, hence there are at least *two* values of j for which X is disjoint from $P^{i_j}_{v_j}$, as required.

The second case is when $|M| \le (2k+1)(4k+2)$. In this case, we show that M satisfies the requirements of Outcome 2. Let $P = p_1, \dots, p_\ell$ be a path as in the statement of the lemma with $\ell \ge 2$, $p_1 = a \in A$, and $p_\ell = v \in K \setminus M$ and suppose that $V(P)$ is disjoint from a k-IP set X. If $p_{\ell'} \in K$ for some $1 \le \ell' < \ell$, then the subpath P' of P from a to $p_{\ell'}$ satisfies the requirements (this includes the case when $a \in K$; Outcome 2 allows that P' has length 0). If $p_{\ell'}$ has more than $2k+1$ neighbors in K for some $1 \le \ell' < \ell$, then $p_{\ell'}$ has a neighbor $v' \in K \setminus (X \cup \{v\})$ (as $|X \cap K| \le 2k$ by Lemma 5). Then the path $P' = p_1, \dots, p_{\ell'}, v'$ satisfies the requirements. Suppose therefore that each vertex $p_{\ell'}$ with $\ell' \in [\ell-1]$ has at most $2k+1$ neighbors in K. This implies that the subtrees corresponding to these vertices do not intersect U and it follows that all these vertices are in the same set C_i. Now the path P shows that $v \in M_i \subseteq M$, a contradiction. ◀

There is a particularly problematic special case that we handle in a separate lemma. It concerns the case when a hole contains a single vertex v from a clique K and the two neighbors of v are in W. It may seem like an easy, degenerate case (after all, rerouting to avoid v means finding another common neighbor of the two neighbors of w in W), but actually it is relatively complicated to find a replacement of v without introducing unwanted adjacencies.

▶ **Lemma 11.** *Let G be a graph with two vertices w_x and w_y such that $G - \{w_x, w_y\}$ is chordal, let K be a clique in $G - \{w_x, w_y\}$, and let k be an integer. In polynomial time, we can find a set $M \subseteq K$ of at most $(2k+4)(2k+1)$ vertices such that the following holds: If X is a k-IP set, and H is a hole disjoint from X and with $V(H) \cap W = \{w_x, w_y\}$ such that H has a vertex $v \in K \setminus M$ adjacent to both w_x and w_y, then there is a hole H' disjoint from $X \cup \{v\}$.*

Proof. Let us consider a subtree representation of the chordal graph G over the tree T. For every node $x \in V(T)$, let us denote by bag B_x the set of those vertices whose subtrees contain x. Let us assume that T is a rooted at a node r and that subtree T_u for every $u \in K$ contains r. For a node x of T, let us define $C_x \subseteq V(G) \setminus \{w_1, w_2\}$ the following way: a vertex u is in C_x if the subtree T_u of u is fully contained in the subtree of T rooted at x. Observe that the neighborhood of C_x (in $G - \{w_1, w_2\}$) is a clique: if u has a neighbor in C_x, but is not itself in C_x, then T_u has to contain the parent node of x.

We say that path P is a *good path* if it is either a path of length 0 consisting a single vertex adjacent to both w_x and w_y, or a path of length at least 1 where the unique vertex adjacent to w_x is one of the endpoints, and the unique vertex adjacent to w_y is the other endpoint. Observe that if P_1 and P_2 are two good paths that have no adjacent vertices, then P_1, P_2, w_x, w_y together form a hole of length at least 4.

Let Z be the set of nodes of T with the following property: a node x is in Z if $G[C_x]$ contains a good path. By definition, Z induces a subtree of T rooted at r: if a node is in Z, then all its ancestors are also in Z. Let $\ell_1, \ldots, \ell_t$ be the leaves of Z.

Let us consider first the case when $T[Z]$ has at least $2k + 5$ leaves. We claim that in this case returning $M = \emptyset$ is a valid answer. As the sets $C_{\ell_1}, \ldots, C_{\ell_t}$ are disjoint, nonadjacent, and the neighborhood of each of them is a clique, Lemma 8 implies that the $(k + 1)$-IP set $X \cup \{v\}$ can intersect at most $2k + 3$ of the sets $C_{\ell_1}, \ldots, C_{\ell_t}$. This means that there are at least two such sets that are disjoint from X; assume, without loss of generality that $X \cup \{v\}$ is disjoint from C_{ℓ_1} and C_{ℓ_2}. By assumption, there are two good paths P_i for $i = 1, 2$ such that P_i is in $G[C_{\ell_i}]$. As there are no edges between P_1 and P_2, we have that P_1, P_2, w_x, w_y together form a hole H' that is disjoint from $X \cup \{v\}$.

Assume therefore that $t \leq 2k + 4$. We construct M using the following procedure. For every $x \in Z$, let M_x contain the vertices $u \in K$ with the property that u is adjacent to both w_1 and w_2, and moreover T_u does not contain x. Observe that $M_x \subseteq M_y$ if x is an ancestor of y. Set $M = \emptyset$ initially. Let us consider the nodes of Z in a top down order, i.e., in a nondecreasing ordering by the distance from the root r. When considering $x \in Z$, extend M using vertices of M_x until either $|M \cap M_x| \geq 2k + 1$ or $M_x \subseteq M$. This completes the definition of M.

We claim that $|M| \leq (2k + 4)(2k + 1)$. Let the weight h_x be the number of vertices added to M when considering node x. We claim that the total weight of a node x and all its ancestors is at most $(2k + 1)$. To prove this, consider a node x such that the total weight of its proper ancestors is $h < 2k + 1$, but $h + h_x > 2k + 1$. Observe that the h vertices added to M by the proper ancestors all appear in M_x and hence we should have added only $2k + 1 - h < h_x$ new vertices of M_x when considering node x, a contradiction. In particular, the statement holds for the leaves of $G[Z]$, hence it follows that the total weight (i.e., the size of M) is at most $t(2k + 1) \leq (2k + 4)(2k + 1)$.

It remains to show that M satisfies the statement of the lemma. Observe that if we remove v, w_x, w_y from the hole H, then what remains is a good path P. Let us choose x to be node of T such that C_x contains P and x has maximum distance to the root r. This means that the bag B_x contains a vertex of P, which also implies that T_v cannot contain x, that is, $v \in M_x$. If $|M_x| \leq 2k + 1$, then the construction of M ensures $M_x \subseteq M$, contradicting $v \in K \setminus M$. If $|M_x| \geq 2k + 2$, then by Lemma 7, M_x contains a vertex v' disjoint from the k-IP set X and different from v. As P is in C_x and $v' \in M_x$, vertex v' is not adjacent to any vertex of P. Thus replacing v with v' in the hole H results in a hole H' that is disjoint from $X \cup \{v\}$. ◀

We are now ready to prove the main result of the section, Lemma 9.

Proof (of Lemma 9). The set $S = V(G) \setminus W$ trivially satisfies Requirement 1. We show that if S satisfies Requirement 1 and has a large clique violating Requirement 2, then we can remove a vertex from S in a way that Requirement 1 remains satisfied. After repeated applications of this argument, we eventually arrive to a set S that satisfies both Requirements 1 and 2.

Suppose that S satisfies Requirement 1, but $G[S]$ has a clique K of size greater than $12(|W|+2)^4$ (as $G[S]$ is chordal, such a clique can be found in polynomial time). We define a set $M \subseteq K$ by invoking the procedure of Lemma 10 with different values for the parameters (G, K, A, k) and then argue that removing from S any vertex $v \in K \setminus M$ does not violate Requirement 1. We need some definitions first. For $w \in W$, let A_w be the neighborhood of w in S. For $w_1, w_2 \in W$, let $A_{w_1,w_2} = A_{w_1} \cap A_{w_2}$ and $A_{w_1,\overline{w_2}} = A_{w_1} \setminus A_{w_2}$. Let $K_{w_1} = K \cap A_{w_1}$, $K_{\overline{w_1}} = K \setminus A_{w_1}$, $K_{w_1,w_2} = K \cap A_{w_1,w_2}$, and $K_{w_1,\overline{w_2}} = K \cap A_{w_1,\overline{w_2}}$.

The set M is the defined the following way. We invoke the algorithm of Lemma 10 with various graphs and sets as listed below.

1. For every $w \in W$, we invoke the procedure with $(G[S], K_{\overline{w}}, A_w, |W|+1)$ and we let $\mathcal{P}_w$ be the set of paths in case of Outcome 1 and M_w be the resulting set in case of Outcome 2.
2. For every $w_1, w_2 \in W$ with $w_1 \neq w_2$, we invoke the procedure with $(G[S \setminus A_{w_2}], K_{\overline{w_2}}, A_{w_1,\overline{w_2}}, |W|+1)$ and we let $\mathcal{P}_{w_1,\overline{w_2}}$ be the set of paths in case of Outcome 1 and $M_{w_1,\overline{w_2}}$ be the resulting set in case of Outcome 2.

The set M is defined to be the union of all these sets M_w and $M_{w_1,\overline{w_2}}$ (for all the values w and $(w_1, \overline{w_2})$ for which the algorithm of Lemma 10 terminated with Outcome 2). Additionally, for every distinct $w_1, w_2 \in W$, we extend M the following way:

1. We put arbitrarily $2|W|+1$ vertices of $K_{w_1,\overline{w_2}}$ into M or all such vertices, if fewer than $2|W|+1$ such vertices exist.
2. We put arbitrarily $2|W|+1$ vertices of K_{w_1,w_2} into M or all such v
3. We extend M with the set returned by the algorithm of Lemma 11 for $G[S \cup \{w_1, w_2\}]$, the clique K, and $k = |W|$.

To bound the size of M, let us observe that we invoke the algorithm of Lemma 10 exactly $|W|+|W|(|W|-1)$ times, each time with parameter $k = |W|+1$. As each call returns a set of size at most $(2k+1)(4k+2)$, the total number of vertices in M obtained this way is at most $8(|W|+2)^4$. Additionally, we include at most $|W|^2(2(2|W|+1)+(2|W|+4)(2|W|+1)) \leq 4(|W|+2)^4$ vertices into M, resulting in at most $12(|W|+2)^4$ vertices in total. We assumed that K has size greater than $12(|W|+2)^4$, hence there exists at least one vertex in $K \setminus M$.

The rest of the proof is devoted to showing that Requirement 1 remains satisfied after removing a vertex $v \in K \setminus M$ from S. If G has no k pairwise disjoint holes, then there is nothing to show; otherwise, as let us fix a collection $\mathcal{H}$ of k pairwise disjoint holes such that every vertex of these holes is in $S \cup W$, except possibly some of the special vertices. Let us choose $\mathcal{H}$ such that the total number of vertices used from W is minimized.

If no hole goes through vertex v, then we are done. Otherwise, let $H \in \mathcal{H}$ be the hole containing v. We may assume that v is not a special endpoint of H, otherwise $S \setminus \{v\}$ still satisfies Requirement 1. Let X be the union of the vertices in $V(G) \setminus W$ used by the holes in $\mathcal{H} \setminus \{H\}$. It is clear that X is a $|W|$-IP set: each hole uses at least one vertex of W and the vertices of W split these holes into a collection of at most $|W|$ induced paths. It also follows that $X \cup \{v\}$ is a $(|W|+1)$-IP set.

As $G - W$ is chordal, H contains at least one vertex of W. The following claim shows that in the definition of M above, the set M_w was defined for every vertex $w \in V(H) \cap W$ (and the same is true for $M_{w,\overline{w'}}$ for any $w' \in W$).

▷ Claim 12. For every $w \in V(H) \cap W$,

1. the set M_w is defined,
2. the set $M_{w,\overline{w'}}$ is defined for every $w' \in W$ with $w' \neq w$.

Proof. If M_w was not defined, then the algorithm of Lemma 10 terminated with Outcome 1 on input $(G[S], K_{\overline{w}}, A_w, |W|+1)$, resulting in a set $\mathcal{P}_w$ of paths satisfying the requirements of Outcome 1. This means that there are two paths $P_1, P_2 \in \mathcal{P}_w$ of length at least one that are disjoint from the $(|W|+1)$-IP set $X \cup \{v\}$. For $i = 1, 2$, let $a_i \in A_w$ and $z_i \in K$ be the endpoints of P_i. Now z_1 and z_2 either coincide or are adjacent in the clique K, thus concatenating them gives a walk that contains an $a_1 - a_2$ simple path P. From the conditions on P_1 and P_2, we also know that a_1 and a_2 are independent and the internal vertices of P are not in A_w, that is, not adjacent to w. Thus Lemma 5 shows that w and P form a hole H' of length at least 4 disjoint from $X \cup \{v\}$ and fully contained in $S \cup W$. Replacing H with H' in $\mathcal{H}$ would give a collection of k holes satisfying Requirement 1, even if v is removed from S, what we wanted to show.

In an analogous way, we can show that if the algorithm of Lemma 10 terminated with Outcome 1 on input $(G[S \setminus A_{w'}], K_{\overline{w'}}, A_{w,\overline{w'}}, |W|+1)$, then there is a path of length at least 2 connecting two independent vertices of $A_{w,\overline{w'}}$ in $G[S \setminus A_w]$ that is disjoint from $X \cup \{v\}$, forming a hole with w. This proves the second statement. ◁

The set $V(H) \setminus W$ induces a set of at least one and at most $|W|$ induced paths in $G - W$ (a path can consists of only a single vertex). Let P be the subpath of H that contains v. Fixing an orientation of H, let $w_x, w_y \in W$ be the previous and next vertices of H (note that $|V(H) \cap W| = 1$ if and only if $w_x = w_y$). We try to reroute P to obtain a path P' that avoids v. Then we replace P with P' in the hole H and try to obtain a hole H' that avoids v, showing that there is a collection of k disjoint holes that satisfies Requirement 1 even if v is removed from S. For this, we need to ensure that the new path P' is independent from the rest of H in a certain way. Thus there are two challenges here: finding the rerouted path P' that avoids v and ensuring that replacing P with P' results in hole.

The rest of the proof depends on the size $V(H) \cap W$. The most generic case is when $V(H) \cap W$ has at least 3 vertices. In this case, we can use that H visits a third vertex $w_z \in H$ different w_x, w_y and P is not adjacent to P; then it is sufficient to ensure that P' is also not adjacent to w_y. The case $|V(H) \cap W| = 2$ is similar, but then we need different arguments to ensure that H' is a hole. In the case $|V(H) \cap W| = 1$, an additional complication is that $w_x = w_y$, hence the endpoints of P' need to be nonadjacent.

Case A: $|V(H) \cap W| \geq 3$. Then $w_x \neq w_y$ and there is at least one other $w_z \in V(H) \cap W$ different from w_x and w_y. Our goal is to show that there is an $A_{w_x,\overline{w_z}} - A_{w_y,\overline{w_z}}$ path P' in $G[S \setminus A_{w_z}]$ that is disjoint from $X \cup \{v\}$. Assuming there is such a path P', let u_1 and u_2 be the two neighbors of w_z on the hole H (note that $u_1, u_2 \notin P$). Replacing P with P' in the hole H shows that that there is a $u_1 - u_2$ walk whose internal vertices are not adjacent to w_z (here we use that the endpoints of P' are adjacent to w_x and w_y, respectively, and the vertices of P' are not adjacent to w_z). Thus by Lemma 5, there is a hole disjoint from $X \cup \{v\}$.

The path P' is constructed as follows. We will construct two paths P'_x and P'_y in $G[S \setminus A_{w_z}]$ such that P'_x and P'_y are $A_{w_x,\overline{w_z}} - K_{\overline{w_z}}$ and $A_{w_y,\overline{w_z}} - K_{\overline{w_z}}$ paths, respectively, and they are both disjoint from $X \cup \{v\}$. As any two vertices of $K_{\overline{w_z}}$ are adjacent, the concatenation of the two paths gives a walk that can be simplified to the required path P'.

The path P can be split into a path P_x going from a vertex $v_x \in A_{w_x,\overline{w_z}}$ to v, and into a path P_y going from a vertex $v_y \in A_{w_y,\overline{w_z}}$ to v (now one or both of these paths can be of length 0). We show how to construct P'_x; the construction of P'_y is analogous. If P_x has length at least one, then we use that, by Claim 12, Outcome 2 was the result of applying Lemma 10 when defining $M_{w_x,\overline{w_z}}$. As P_x is a path from v_x to $v \in K_{\overline{w_z}} \setminus M \subseteq K_{\overline{w_z}} \setminus M_{w_x,\overline{w_z}}$ and it is disjoint from X, Outcome 2 guarantees the existence of a path P'_x from v_x to $K_{\overline{w_z}}$ that is disjoint from $X \cup \{v\}$. If P_x has length 0, then $v_x = v$ is in $K_{w_x,\overline{w_z}}$. When defining the set M, we tried to put $2|W|+1$ vertices of $K_{w_x,\overline{w_z}}$ into M. As v was not put into this set, we have that M contains $2|W|+1$ other vertices of $K_{w_x,\overline{w_z}}$. As the $|W|$-IP set X can contain at most $2|W|$ vertices of $K_{\overline{w_z}}$ (Lemma 7), there is a vertex $v' \in K_{w_x,\overline{w_z}} \setminus X$ different from v. Now the path P'_x of length 0 consisting of only v' satisfies the requirements. Path P'_y can be obtained in a similar way, completing our proof for the existence of the path P'.

Case B: $|V(H) \cap W| = 2$. In this case, $H - W$ consists of either only the path P, or two paths P and P^*. Note that if $H - W$ consists of only P, then the two vertices in $w_x, w_y \in V(H) \cap W$ are adjacent and P has length at least two. Let us first handle the case when P has length 0, i.e., it consist of a single vertex v adjacent to both w_x and w_y. Then the fact that M includes the set returned by Lemma 11 for graph $G[S \cup \{w_x, w_y\}]$, clique K, $k = |W|$ implies that there is a hole H' disjoint from the $|W|$-IP set X and from v.

In the following, we assume that P has length at least 1 (which in particular implies that P has no vertex adjacent to both w_x and w_y). If P^* exists, then we claim that no vertex of P^* is adjacent to a vertex $u \in K \setminus (X \cup K_{w_x,w_y})$. Otherwise, without loss of generality, assume that u is not adjacent to w_x (the other case being symmetric). Let v_x be an endpoint of P adjacent to w_x, and consider the subpath Q of P from v to v_x. Let v^*_x be an endpoint of P^* adjacent to w_x and consider the subpath Q^* of P^* from v^*_x to a vertex u^* that is adjacent to u. Now v_x and v^*_x are nonadjacent neighbors of w_x (because the paths P and P^* are not adjacent) and the walk Q^*u^*uvQ goes from v^*_x to v_x. Note that no internal vertex of this walk is adjacent to w_x (as we assumed that this is true for u). Therefore, there is a hole H' that is disjoint from the holes in $\mathcal{H} \setminus \{H\}$ (the only vertex possibly used by H' that is not used by H is $u \in K \setminus X$) and uses only one vertex of W, contradicting the minimal choice of $\mathcal{H}$.

If P has length at least 1, then it has an endpoint $v_x \in A_{w_x,\overline{w_y}}$ and an endpoint $v_y \in A_{w_y,\overline{w_x}}$. Then we proceed very similarly to Case A above. The path P can be split into a path P_x going from vertex $v_x \in A_{w_x,\overline{w_y}}$ to v, and into a path P_y going from vertex $v_y \in A_{w_y,\overline{w_y}}$ to v (one, but not both, of these paths can be of length 0). We construct two paths P'_x and P'_y in G such that P'_x and P'_y are $A_{w_x,\overline{w_y}} - K$ and $A_{w_y,\overline{w_x}} - K$ paths, respectively, and they are both disjoint from $X \cup \{v\}$. If P_x has length at least 1, then the definition of M_{w_x} obtained by Outcome 2 of Lemma 10 (which exists by Claim 12) shows that a path P'_x disjoint from $X \cup \{v\}$ exists, and moreover any vertex of P'_x not in $V(P_x)$ is in $K_{\overline{w_x}}$. If P_x is only a single vertex, $v_x = v$ and the definition of M includes at least $2|W|+1$ vertices from $K_{w_x,\overline{w_y}}$ and there exists a vertex $v' \in K_{w_x,\overline{w_y}} \setminus (X \cup \{v\})$. Path P'_y can be constructed similarly. Putting together these two paths gives an $A_{w_x,\overline{w_y}} - A_{w_y,\overline{w_x}}$ walk Q. Observe that no vertex of Q is in A_{w_x,w_y}: there is no such vertex in $V(P_x) \cup V(P_y)$ and the new vertices we may introduce when defining P'_x or P'_y came from $K_{\overline{w_y}}$ or from $K_{\overline{w_x}}$, hence they cannot be in A_{w_x,w_y} either. Therefore, walk Q has a simple $A_{w_x,\overline{w_y}} - A_{w_y,\overline{w_x}}$ subpath P' of length at least 1 whose internal vertices are disjoint from A_{w_x} and A_{w_y}. As P' was constructed in such a way that any vertex of it not in $V(P)$ is in the clique $K \setminus (X \cup K_{w_x,w_y})$, our claim in the previous paragraph shows that P^* (if exists) is not adjacent to P'. Thus replacing w_xPw_y with $w_xP'w_y$ in the hole H shows the existence of a hole H' disjoint from $X \cup \{v\}$.

Case C: $|V(H) \cap W| = 1$. Let $V(H) \cap W = \{w\}$. We again proceed similarly as in Case A, but we use the fact v is not a special endpoint of H, that is, v is not an endpoint of P. The path P can be split into a path P_x going from a vertex $v_x \in A_w$ to v, and into a path P_y going from a vertex $v_y \in A_w$ to v; both paths have length at least 1 (as v is not a special endpoint) and v_x and v_y are not adjacent. We will construct a $v_x - K$ path P'_x and a $v_y - K$ path P'_y in G that are disjoint from $X \cup \{v\}$ and moreover v_x and v_y are the only vertices of A_w on these paths. To construct P'_x, consider the set M_w, which was defined by Outcome 2 of Lemma 10 applied on $(G[S], K_{\overline{w}}, A_w, |W|+1)$ (by Claim 12, the set M_w is defined). As P_x has length at least 1 and $v \notin A_w$, the assumption $v \notin K \setminus M_w$ implies that there is a path P'_x from v_x to a vertex of K that is disjoint from $X \cup \{v\}$ and has exactly one vertex in A_w (namely, v_x). Moreover, any vertex of P'_x not in $V(P_x)$ is in $K_{\overline{w}}$, it follows that v_x is the only vertex of P'_x in A_w, as required. Path P'_y can be constructed in a similar way. Then P'_x and P'_y show that existence of a path P' from v_x to v_y with no internal vertices in A_w. Finally, Lemma 5 applied on w and P' shows the existence of a hole H' disjoint from $X \cup \{v\}$. ◀

4 Part 2: Special Hole Packing

Lemma 9 did not manage to fully reduce the treewidth of the Hole Packing instance. To proceed, we introduce a generalization Hole Packing and show that in this generalization it is possible to encode the original instance of Hole Packing in a way that the treewidth is reduced to a function of size of the chordal deletion set W given in the input.

We define the Special Hole Packing problem the following way. The input is a vertex-labeled graph G and two integers $\ell \leq k$. The possible labels of the vertices are 0, i, or i^* for $i \in [\ell]$ (i.e., there are $2\ell + 1$ different labels). The task is to find k pairwise vertex-disjoint holes $H_1, \ldots, H_k$ with the following additional conditions. For $\ell < i \leq k$, every vertex of H_i should have label 0. For $i \in [\ell]$, hole H_i should have exactly one vertex v_i with label i and every other vertex of H_i has label 0. Moreover, if v'_i, v''_i are the neighbors of v_i in H_i, then there should exist a $v'_i - v''_i$ path P_i whose internal vertices have label i^*.

Using Courcelle's Theorem (Theorem 4), it is not difficult to show that Special Hole Packing is FPT with combined parameters k and the treewidth of G: it is routine to describe the problem in Monadic Second Order Logic. A tedious but standard dynamic programming algorithm can also show that running time $2^{\text{poly}(k+\text{tw}(G))} \cdot n$ is also possible.

▶ **Lemma 13.** Special Hole Packing *is FPT parameterized by* $k + \text{tw}(G)$*.*

The final part of the proof is to reduce Hole Packing to Special Hole Packing on a bounded treewidth graph. The main idea is the following. We first use Lemma 9 to mark a set S of vertices in the chordal graph $G - W$. We can assume that if a vertex v is not in $S \cup W$, then it may only be used as a special vertex of a special hole. We treat such vertices v as follows. Suppose that v is adjacent to w_i and represented by a subtree T_v in the clique tree decomposition of $G - W$. Let $a_1, \ldots, a_t$ be the leaves of T_v. Then we "blow up" v: we remove it and for each a_j, we introduce a new vertex v_j that is adjacent to w_i and whose representation in the clique tree decomposition is just the single node a_j. We perform this step for every $v \notin S \cup W$. Using easy transformations of the clique tree, we can ensure that every node of the clique tree is the leaf of at most one subtree, hence we add at most one new vertex at each node. This ensures that the resulting modified graph has bounded treewidth.

There are at least two obvious problems with this transformation. First, vertex v is replaced by multiple vertices $v_1, \ldots, v_t$ and the solution may use more than one of them, effectively using v in more than one hole. However, we can use the Color Coding [2] technique in a straightforward way to enforce that each vertex is used only in one hole as special vertex. The second problem is that the transformation loses information about adjacency. Suppose that we find a solution that contains a special hole H_i that consists of vertex w_i and a path P, where path P connects a newly introduced vertex v_j with a vertex u (that is not adjacent to v_j). Now the orginial vertex v could be adjacent to u in the original graph, hence replacing v_j with v may not give a hole in the original graph. This is the point where the required path P_i on vertices with label i^* comes into play: we introduce these vertices in a way that forces the $v_j - u$ path to "go away" from the tree T_v, making sure that it ends at a vertex u that is not a neighbor of v. More precisely, we introduce a tree-like "scaffolding" with label i^* and then we break this scaffolding in a way that potential paths of label i^* can touch only the leaves of every such tree T_v.

We state the main result of the section as the fixed-parameter tractability of Hole Packing parameterized by the size of a chordal deletion set given in the input.

▶ **Lemma 14.** *Given a graph G, integer k, and vertex set $W \subseteq V(G)$ such that $G - W$ is chordal, the* Hole Packing *instance (G, k) can be solved in time $f(|W|)n^{O(1)}$.*

Proof. Let S be the set given by Lemma 9 and let us fix a hypothetical solution $\mathcal{H}$ of k holes such that every vertex of every hole is in $S \cup W$, except perhaps some of the special vertices. If $w_1, \ldots, w_\ell$ are distinct vertices from W, then we say that $\mathcal{H}$ is *consistent* with the tuple $(w_1, \ldots, w_\ell)$ if each w_i for $i \in [\ell]$ is in a special hole and no other vertex of W is in a special hole; in particular, this implies that there are exactly ℓ special holes. The algorithm first guesses a tuple $(w_1, \ldots, w_\ell)$ with which $\mathcal{H}$ is consistent (as the order of the w_i's do not matter, we have $2^{|W|}$ different possibilities). In the following, let $H_i \in \mathcal{H}$ be the special hole going through w_i (note that each special hole uses exactly one vertex of W hence the H_i's are distinct).

Let $\lambda : V(G) \setminus (W \cup S) \to [\ell]$ be an arbitrary labeling and let X_i be the set of vertices with label i. We say that $\mathcal{H}$ is *consistent* with $(\lambda; w_1, \ldots, w_\ell)$ if it is consistent with $(w_1, \ldots, w_\ell)$ and moreover the special endpoints of H_i are in $X_i \cup S$. Observe that this definition puts a requirement on the labeling of at most 2ℓ vertices. Thus we can use Color Coding [2]: by going through a 2ℓ-perfect family of hash functions of size $2^{O(\ell)} \cdot n$, we can assume that we have a fixed $(\lambda; w_1, \ldots, w_\ell)$ with which the hypothetical solution $\mathcal{H}$ is consistent.

Given G and $(\lambda; w_1, \ldots, w_\ell)$, our goal is to obtain a labeled bounded-treewidth graph G' and invoke the algorithm of Lemma 13 for Special Hole Packing on (G', k). To define G', let us fix a clique tree decomposition of the chordal graph $G - W$. It will be convenient to assume the following extra properties of the clique tree decomposition:

(P1) Every tree T_x has at least two vertices.
(P2) The maximum degree of T is at most 3.
(P3) If u and v are adjacent in T and one of them has degree 3, then $B_u = B_v$.
(P4) Every leaf of every subtree T_x is in a degree-2 node of T.
(P5) Every node of u is the leaf of at most one subtree T_x.

Property (P1) can be achieved by attaching a new leaf to each node u, having the same bag B_u. Properties (P2) and (P3) can be achieved by replacing each node u of degree $d \geq 3$ with a binary tree having d leaves and each node having the same bag B_u. Then the neighbors of u can be connected to the leaves of this tree. This replacement also ensures that every leaf of every subtree T_x is in a node with degree at most 2. Therefore, Property (P4) can be achieved simply by attaching a new node with empty bag to each leaf node of T (to avoid that some T_x has a leaf in a degree-1 node). Property (P5) can be achieved by an appropriate sequence of subdivisions at each edge of T.

To construct G', let us start with $G[S \cup W]$, let us assign label i to w_i and label 0 to every other vertex. For every $i \in [\ell]$, we proceed the following way. We will add new vertices to G' and the way we are describing these new vertices is by adding new subtrees to the clique tree decomposition of $G - W$ (which defines how these new vertices are adjacent to the vertices not in W) and explicitly specifying how the new vertices are adjacent to W. Note that $G' - W$ will be a chordal graph defined by this clique tree decomposition. First, for every edge uv of T, we introduce a new vertex with label i^* whose subtree consists of nodes u and v. Next, for every vertex $x \in X_i$ that is adjacent to w_i, and for every leaf u of T_x, we do the following. By (P1), subtree T_u has at least two vertices, thus u has a unique neighbor v that is in T_x. We remove the vertex with label i^* whose subtree consists of $\{u, v\}$ and introduce a new vertex with label 0 that is adjacent to w_i and whose subtree consists of only $\{u\}$. This completes the description of G'.

We claim that G' has treewidth at most $12(|W|+2)^2+4|W|$. The graph $G[S]$ is a chordal graph with maximum clique size $12(|W|+2)^4$, hence each bag contains at most that many vertices. For every i, we introduce at most 3 new vertices with label i^* in each bag (as (P2) requires that every node of T have degree at most 3). Furthermore, as each node of T contains the leaf of at most one subtree T_x by (P5), we may introduce at most one new vertex with label 0 in each bag. Therefore, $G' - W$ has a clique tree decomposition where every bag has size at most $12(|W|+2)^2+3|W|+1$. This means that $G' - W$ has treewidth at most $12(|W|+2)^2+3|W|$ and hence G' has treewidth at most $12(|W|+2)^2+4|W|$. Therefore, by Lemma 13, we can solve Special Hole Packing for (G', k) in time $f(|W|) \cdot n^{O(1)}$. The following claim shows that in case we find a solution to this Special Hole Packing instance, then it allows us to find k disjoint holes in G.

▷ Claim 15. Given a solution for the Special Hole Packing instance (G', k), we can construct in polynomial time a set of k pairwise vertex-disjoint holes in G.

Proof. Let $H'_1, \ldots, H'_k$ be the solution of the Special Hole Packing instance (G', k). We show first that for $i > \ell$, hole H'_i in G' is a hole in G as well. Recall that for $i > \ell$, hole H'_i contains only vertices of label 0. The only potential problem is that H'_i contains a vertex x^* that do not appear in G, but was added during the construction of G'. But recall that every such vertex x^* was defined by introducing a subtree T_{x^*} containing only a single node u of T and x^* was also made adjacent to a single w_j for some $j \in [\ell]$. Thus the neighbors of x^* is w_j plus a clique, which means that any hole going through x^* has to go through w_j as well. As w_j is the only vertex with label j, we have that H'_j has to contain it and hence hole H'_i for $i > \ell$ does not contain w_j. This shows that H'_i is a hole in G as well.

Consider now the hole H'_i for some $i \in [\ell]$ and the path P_i that connect neighbors x, y of w_i in H'_i and whose internal vertices are labeled i^*. This hole may contain vertices not present in G, but the argument in the previous paragraph shows that there are at most two such vertices: x and y. We show that if these vertices are not present in G, then they can be replaced by vertices in X_i, resulting in a hole H_i of G.

As x and y are not adjacent, there is a path $p_1 p_2 \ldots p_t$ in T with $t \geq 1$ such that p_1 is the only vertex of the path in T_x and p_t is the only vertex of the path in T_y. For every $j \in [t-1]$, the construction of G' involved introducing an i^*-labeled vertex z_j whose subtree is exactly $\{p_j, p_{j+1}\}$. It is clear that the path P_i consists of the vertices $x, z_1, \ldots, z_t, y$: there is no other way of connecting x and y with a simple path whose internal vertices have label i^*. If x is not a vertex of G, then x was introduced in the construction of G' because there is a vertex $x_0 \in X_i$ that is adjacent to w_i and p_1 is the leaf of the subtree T_{x_0}; let us replace x with x_0 in the hole H'_i. Similarly, if y is not part of G, then y can be replaced by a vertex $y_0 \in X_i$ that is adjacent to w_i and whose subtree T_{y_0} contains p_t. We have to verify that the hole H_i obtained by replacing x with x_0 and/or y with y_0 is indeed a hole. First, $V(T_x) \subseteq V(T_{x_0})$ (as T_x contains only node p_1), hence x_0 is adjacent to every neighbor of x. Therefore, we only need to verify that no unwanted new edge appears in H_i after the replacement. The crucial point here is that p_2 is not in T_{x_0}: in that case, we would have removed vertex z_1 during the construction of G'. Thus p_1 is the only node of T_{x_0} on the path $p_1, \ldots, p_t$. Similarly, if y is replaced by y_0, then p_t is the only node of T_{y_0} on this path. It is easy to see that the subtree of every vertex of $V(H'_i) \setminus \{w_i\}$ contains a node from this path and therefore if, e.g., x_0 is adjacent to such a vertex, then already x was adjacent to that. Thus after the replacements, we indeed obtain a hole H_i in G. Performing this step for every $i \in [\ell]$ gives a set $H_1, \ldots, H_\ell, H'_{\ell+1}, \ldots, H'_k$ of holes. The disjointness of these holes follow from the disjointness of $H'_1, \ldots, H'_k$ and from the fact that $V(H_i) \setminus V(H'_i)$ is in X_i, and these vertices cannot be used by any hole other than H_i. ◁

Conversely, the (contraposition of the) following claim shows that if the answer to the Special Hole Packing instance is no, then we know that there is no set of disjoint holes consistent with our current choice of $(\lambda; w_1, \ldots, w_\ell)$.

▷ Claim 16. If G has a set $\mathcal{H}$ of pairwise disjoint holes consistent with $(\lambda; w_1, \ldots, w_\ell)$, then Special Hole Packing for (G', k) has a solution.

Proof. Let $H_1, \ldots, H_k$ be disjoint holes in G such that H_i for $i \in [\ell]$ is a special hole going through vertex w_i. For special hole H_i where vertices x and y are the neighbors of w_i, we define the *gap size* of H_i to be the distance of T_x and T_y in T. As x and y are not adjacent, the gap size is positive. Let us choose $H_1, \ldots, H_k$ such that the sum of gap sizes is minimum possible.

For $i > \ell$, the vertices of H_i are contained in $S \cup W$, hence they are also holes in G' as well with every vertex having label 0. Consider now the special hole H_i for $i \in [\ell]$ and suppose it has the form $w_i x x' P y' y$ where P is an $x' - y'$ path (with $x' = y'$ if H_i has length 4).

As x and y are not adjacent, there is a path $p_1 p_2 \ldots p_t$ in T with $t \geq 1$ such that p_1 is the only vertex of the path in T_x and p_t is the only vertex of the path in T_y. Observe that this means that the gap size of H_i is exactly $t-1$. It easy to see that every bag B_{p_i} for $i \in [t]$ contains at least one vertex of the path P. By (P4), the leaves are in degree-2 nodes, hence there is a unique vertex p_0 before p_1 and a unique vertex p_{t+1} after p_t. For every $j \in [t-1]$, the construction of G' involved introducing an i^*-labeled vertex z_j whose subtree is exactly $\{p_j, p_{j+1}\}$. We argue that none of these vertices z_j were removed during the construction of G'. Recall that z_j was removed if there was a vertex $q \in X_i$ that is adjacent to w_i and either T_q has a leaf in p_{j+1} and T_q contains p_j, or T_q has a leaf in p_j and T_q contains p_{j+1}. We show that if a z_j was removed during the construction of G', then H_i can be replaced with another hole that has strictly smaller gap size and is still disjoint from the rest of the holes. This would contradict the minimality of the choice of $\mathcal{H}$.

Let α be the largest integer $\leq t$ such that there is a vertex in $x^* \in X_i \cup \{x\}$ whose subtree has a leaf in p_α and contains $p_{\alpha-1}$. Vertex x shows that α is well-defined and at least 1. Furthermore, vertex y shows that α cannot be t: by (P5), node p_t is the leaf of only T_y and this subtree does not contain p_{t-1}. Let $\beta \geq \alpha$ be the smallest integer such that there is a vertex in $y^* \in X_i \cup \{y\}$ whose subtree has a leaf in p_β and contains $p_{\beta+1}$. Vertex y shows that β is well-defined and $\beta \leq t$. Furthermore, we have that $\beta \neq \alpha$, as otherwise both x^* and y^* would have a leaf at the same node (and they are distinct vertices). Now $\beta > \alpha$ implies that x^* and y^* are not adjacent. It can be also observed that vertex z_j with $\alpha \leq j \leq \beta - 1$ was not removed: removing it because of a subtree with leaf in p_{j+1} and containing p_j would contradict the maximality of α; removing it because of a subtree with leaf in p_j and containing p_{j+1} would violate the minimality of β. Therefore, if $\alpha = 1$ and $\beta = t$, then none of $z_1, \ldots, z_{t-1}$ was removed, what we wanted to show. Otherwise, there is an $x^* - y^*$ path whose internal vertices are in $V(P)$ (as every bag $B_{p_\alpha}, \ldots, B_{p_\beta}$ contains a vertex of P). Then Lemma 5 implies that there is a hole H_i^* going through x^*, w_i, and y^*. This hole is disjoint from the other holes in $\mathcal{H} \setminus \{H_i\}$: vertices in $V(H_i^*) \setminus V(H_i)$ are from X_i, and consistency of $\mathcal{H}$ with $(\lambda; w_1, \ldots, w_\ell)$ means that only H_i could use vertices from X_i. Now the gap size of H_i^* is $\beta - \alpha < t - 1$, strictly smaller than the gap size of H_i, contradicting the minimal choice of $\mathcal{H}$. This completes the proof that all of the vertices z_1, $\ldots$, z_{t-1} are in G'.

Let us show now the existence of a hole H_i' required by the Special Hole Packing problem. If $x, y \in S$, then they are present in G' with label 0, hence H_i is a hole in G' with every vertex except w_i having label 0. Then $P_i = xz_1 \ldots z_{t-1}y$ is a path with internal vertices labeled i^*, as required. Otherwise, suppose that $x \notin S$. As $\mathcal{H}$ is consistent with $(\lambda; w_1, \ldots, w_\ell)$, this is only possible if $x \in X_i$. As x is adjacent to w_i and p_1 is a leaf of T_x, we have introduced a vertex x^* that is adjacent to w_i and whose subtree consists of only p_1. Let us observe that x^* is adjacent to x': this follows from the fact that the induced path $xx'Py'y$ connects x and y, hence the subtree $T_{x'}$ should contain a node of the component of $T - V(T_x)$ that contains T_y. As $V(T_{x^*}) \subseteq V(T_x)$, vertex x^* cannot be adjacent to any vertex that is not a neighbor of x. Let us replace x with x^*; in a similar way, if $y \notin S$, then we can replace it with an appropriate vertex y^* that is adjacent to w_i and y'. Then the resulting hole H_i' is a hole in G' where every vertex except w_i has label 0. Furthermore, the two neighbors of w_i in H_i' can be connected by a path P_i whose internal vertices are $z_1, \ldots,$ z_{t-1}, as required in the definition of Special Hole Packing. ◁

In summary, we can solve Hole Packing the following way. We enumerate every possibility for $(w_1, \ldots, w_\ell)$ and every mapping λ in a 2ℓ-perfect family of hash functions. For each choice of $(\lambda; w_1, \ldots, w_\ell)$, we construct the graph G' and solve the Special Hole Packing instance (G', k) using the algorithm of Lemma 13. If it returns a solution, then Claim 15 allows us to turn this into a solution of Hole Packing. The correctness of the algorithm follows from the fact if there is a set $\mathcal{H}$ of k pairwise disjoint holes, then at some point we reach a tuple $(\lambda; w_1, \ldots, w_\ell)$ with which $\mathcal{H}$ is consistent. At this point, Claim 16 shows that the Special Hole Packing instance (G', k) has a solution, and hence our algorithm indeed returns a solution for Hole Packing. ◀

To prove our main result Theorem 3, let us invoke the algorithm of Theorem 2 on the Hole Packing instance (G, k). If it returns a set of $k + 1$ disjoint holes, then we are done. Otherwise, we can assume that we have a set W of $O(k^2 \log k)$ vertices such that $G - W$ is chordal. Then we can use Lemma 14 to solve the problem in time $f(|W|)n^{O(1)} = f'(k)n^{O(1)}$.

5 A difficult situation

The set S computed by Lemma 9 does not necessarily cover the special vertices of the special holes. If we could ensure that set S covers every vertex of the solution, then we could immediately apply Courcelle's Theorem and we would not need the arguments in Section 4. This raises the question whether we could improve the proof of Lemma 9 in such a way.

Of course, a solution can use at most $2k$ vertices from a clique, so a large clique certainly has a vertex v not needed for a solution. Therefore, technically speaking, we can prove a variant of Lemma 9 without the extra condition for the special vertices: using our algorithm for Hole Packing, we can find a set k disjoint holes and we can set S to be the vertices of these holes. A better question is whether we can prove the inductive rerouting argument in the proof of Lemma 9 without the extra provision for the special vertices. Notice that the proof of Lemma 9 shows the following: if a hole of the solution goes through a vertex $v \notin S$ of a large clique, then we can reroute that hole *without modifying any of the other holes.* Therefore, the question is whether we can find a vertex in a large clique that is irrelevant for the existence of a hole, even after the deletion of a set of other holes.

▶ **Question 17.** *Let G be a graph and W a set of vertices such that $G - W$ is chordal and let K be a clique in $G - W$. We say that $v \in K$ is* irrelevant *if whenever $\mathcal{H}$ is a collection of disjoint holes*[1] *in G such that $G - (\bigcup_{H \in \mathcal{H}} V(H))$ contains a hole, then $G - (\bigcup_{H \in \mathcal{H}} V(H) \cup \{v\})$ contains a hole as well. Is there a function f such that every clique K of $G - W$ having size at least $f(|W|)$ contains an irrelevant vertex?*

We construct a simple example that gives a negative answer to this question. Therefore, when trying to declare a vertex v as irrelevant, we cannot argue just by rerouting the hole going through v in the solution: we may need to reroute some other holes of the solution as well. This shows that a version of Lemma 9 without the provision for the special vertices would require a proof of very different flavor.

We define first a chordal graph G_0 the following way. Let the tree T contain nodes x_0, $\dots$, x_{n+1} forming a path and let y_i be a degree-1 neighbor of x_i for $i \in [n]$. We define chordal graph G_0 as the intersection graph of subtrees of T:

- For $i \in [n]$, vertex $a_i \in V(G_0)$ corresponds to a subtree with nodes $\{x_{i-1}, x_i, x_{i+1}, y_i\}$.
- For $3 \le i \le n-2$, vertex $b_i \in V(G_0)$ corresponds to a subtree with nodes $\bigcup_{j \in [n]} \{x_j, y_j\} \setminus \{y_i\}$.
- For $i \in [n]$, vertex $c_i \in V(G_0)$ corresponds to a subtree with node $\{y_i\}$.

Observe that the a_i's form a path P, the b_i's form a clique K and the c_i's form an independent set I. Graph G is defined by adding three new vertices w_1, w_2, w_3 to G_0, making w_1 and w_2 adjacent to I, and making w_3 adjacent to $K \cup I$.

Let us choose an arbitrary $3 \le i \le n-2$. Consider the holes $H_1 = w_1 c_1 a_1 a_2 \dots a_{i-1} c_{i-1} w_1$ and $H_2 = w_2 c_{i+1} a_{i+1} a_{i+2} \dots a_n c_n w_2$. We claim that $H_3 = w_3 c_i a_i b_i w_3$ is the only hole in $G - (V(H_1) \cup V(H_2))$, showing that b_i is not an irrelevant vertex. As $G - \{w_1, w_2, w_3\}$ is chordal, such a hole has to go through a w_3 and then contain two nonadjacent neighbors of w_3. As K is a clique, this means that hole H contains c_j for some $j \in [n]$. Consider the neighbor of c_j in H different from w_3. This vertex cannot be from K (as it cannot be a neighbor of w_3), hence a_j is the only possible neighbor of c_j. Hole H_1 uses the vertices a_1, $\dots$, a_{i-1}, while hole H_2 uses the vertices a_{i+1}, $\dots$, a_n, which leaves only a_i, and $i = j$ follows. Therefore, hole H contains vertices $w_3 c_i a_i$, which can be completed to a hole only by vertex b_i (the only vertex of K not adjacent to c_i), as claimed. As this argument holds for every $3 \le i \le n-2$, none of the $n-4$ vertices of the clique K are irrelevant. The construction is valid for arbitrary large n and we have $|W| = 3$, which rules out the possibility that the function f of Question 17 depending only on $|W|$ exists.

References

1 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.

2 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.

[1] Note that $\mathcal{H}$ can contain at most $|W|$ holes.

3 Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernel bounds for path and cycle problems. *Theor. Comput. Sci.*, 511:117–136, 2013. doi:10.1016/j.tcs.2012.09.006.

4 Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.*, 58(4):171–176, 1996. doi:10.1016/0020-0190(96)00050-6.

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

6 Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5):21:1–21:19, 2008. doi:10.1145/1411509.1411511.

7 Bruno Courcelle. Graph rewriting: an algebraic and logic approach. In *Handbook of theoretical computer science, Vol. B*, pages 193–242. Elsevier, Amsterdam, 1990.

8 Christophe Crespelle, Pål Grønås Drange, Fedor V. Fomin, and Petr A. Golovach. A survey of parameterized algorithms and the complexity of edge modification. *CoRR*, abs/2001.06867, 2020. arXiv:2001.06867.

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

10 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

11 P. Erdős and L. Pósa. On independent circuits contained in a graph. *Canad. J. Math.*, 17:347–352, 1965.

12 Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. On the parameterized complexity of graph modification to first-order logic properties. *Theory Comput. Syst.*, 64(2):251–271, 2020. doi:10.1007/s00224-019-09938-8.

13 Fedor V. Fomin, Saket Saurabh, and Neeldhara Misra. Graph modification problems: A modern perspective. In Jianxin Wang and Chee-Keng Yap, editors, *Frontiers in Algorithmics - 9th International Workshop, FAW 2015, Guilin, China, July 3-5, 2015, Proceedings*, volume 9130 of *Lecture Notes in Computer Science*, pages 3–6. Springer, 2015. doi:10.1007/978-3-319-19647-3_1.

14 Martin Charles Golumbic. *Algorithmic graph theory and perfect graphs*. Academic Press, New York, 1980.

15 Jens Gramm, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Automated generation of search tree algorithms for hard graph modification problems. *Algorithmica*, 39(4):321–347, 2004. doi:10.1007/s00453-004-1090-5.

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

17 Naonori Kakimura, Ken-ichi Kawarabayashi, and Dániel Marx. Packing cycles through prescribed vertices. *J. Comb. Theory, Ser. B*, 101(5):378–381, 2011. doi:10.1016/j.jctb.2011.03.004.

18 Ken-ichi Kawarabayashi and Bruce A. Reed. Odd cycle packing. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 695–704. ACM, 2010. doi:10.1145/1806689.1806785.

19 Ken-ichi Kawarabayashi, Bruce A. Reed, and Paul Wollan. The graph minor algorithm with parity conditions. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 27–36. IEEE Computer Society, 2011. doi:10.1109/FOCS.2011.52.

20 Eun Jung Kim and O-joung Kwon. Erdős-pósa property of chordless cycles and its applications. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1665–1684. SIAM, 2018. doi:10.1137/1.9781611975031.109.

21 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is np-complete. *J. Comput. Syst. Sci.*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4.

22 Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010. doi:10.1007/s00453-008-9233-8.

23 M. Pontecorvi and Paul Wollan. Disjoint cycles intersecting a set of vertices. *J. Comb. Theory, Ser. B*, 102(5):1134–1141, 2012. doi:10.1016/j.jctb.2012.05.004.

24 Bruce A. Reed. Mangoes and blueberries. *Combinatorica*, 19(2):267–296, 1999. doi:10.1007/s004930050056.

25 Bruce A. Reed, Neil Robertson, Paul D. Seymour, and Robin Thomas. Packing directed circuits. *Combinatorica*, 16(4):535–554, 1996. doi:10.1007/BF01271272.

26 Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004. doi:10.1016/j.orl.2003.10.009.

27 Donald J. Rose, Robert Endre Tarjan, and George S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5(2):266–283, 1976. doi:10.1137/0205021.

28 Aleksandrs Slivkins. Parameterized tractability of edge-disjoint paths on directed acyclic graphs. *SIAM J. Discrete Math.*, 24(1):146–157, 2010. doi:10.1137/070697781.

29 Stéphan Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Trans. Algorithms*, 6(2):32:1–32:8, 2010. doi:10.1145/1721837.1721848.

30 Mihalis Yannakakis. Edge-deletion problems. *SIAM J. Comput.*, 10(2):297–309, 1981. doi:10.1137/0210021.

31 Mihalis Yannakakis. Node-deletion problems on bipartite graphs. *SIAM J. Comput.*, 10(2):310–327, 1981. doi:10.1137/0210022.