Algorithmica (2010) 57: 747–768
DOI 10.1007/s00453-008-9233-8

# Chordal Deletion is Fixed-Parameter Tractable

Dániel Marx

Received: 20 February 2008 / Accepted: 7 October 2008 / Published online: 30 October 2008

**Abstract** It is known to be NP-hard to decide whether a graph can be made chordal by the deletion of $k$ vertices or by the deletion of $k$ edges. Here we present a uniformly polynomial-time algorithm for both problems: the running time is $f(k) \cdot n^{\alpha}$ for some constant $\alpha$ not depending on $k$ and some $f$ depending only on $k$. For large values of $n$, such an algorithm is much better than trying all the $O(n^k)$ possibilities. Therefore, the chordal deletion problem parameterized by the number $k$ of vertices or edges to be deleted is fixed-parameter tractable. This answers an open question of Cai (Discrete Appl. Math. 127:415–429, 2003).

**Keywords** Chordal graphs · Fixed-parameter tractability · Chordal deletion

## 1 Introduction

A graph is chordal if it does not contain an induced cycle of length greater than 3. It can be decided in linear time whether a graph is chordal [26]. However, it is NP-complete to decide whether a graph can be made chordal by the deletion of $k$ vertices [19], by the deletion of $k$ edges [23], or by the addition of $k$ edges [27] (if $k$ is part of the input).

In this paper we investigate these problems from the parameterized complexity point of view. Parameterized complexity deals with problems where the input has a distinguished part $k$ (usually an integer) called the *parameter*. A parameterized problem is called *fixed-parameter tractable (FPT)* if there is an algorithm with running

Research is supported by the Magyary Zoltán Felsőoktatási Közalapítvány and the Hungarian National Research Fund (OTKA grant 67651).

D. Marx (✉)
Department of Computer Science and Information Theory, Budapest University of Technology and Economics, Budapest 1521, Hungary
e-mail: dmarx@cs.bme.hu

time $f(k) \cdot n^\alpha$, where $f(k)$ is an arbitrary function and $\alpha$ is a positive constant independent of $k$. It turns out that several NP-hard decision problems, such as MINIMUM VERTEX COVER (parameterized by the size $k$ of the vertex cover to be found) and LONGEST PATH (parameterized by the length $k$ of the path), are fixed-parameter tractable. The function $f(k)$ is usually exponential, thus if the parameter $k$ can be arbitrary, then the algorithms are not polynomial (as expected). However, for small fixed values of $k$, fixed-parameter tractable problems have low-degree polynomial algorithms, which are sometimes even practically feasible. The definition of fixed-parameter tractability can be extended in a straightforward way to the case when the input has two parameters $k_1, k_2$. In this case, our aim is to find an algorithm with running time $f(k_1, k_2) \cdot n^\alpha$. For more background, the reader is referred to the monograph of Downey and Fellows [8] or to the recent book of Flum and Grohe [9].

If $k$ is a fixed constant, then the three chordal deletion/completion problems can be solved in polynomial time by exhaustive search. For example, in the edge completion problem we can try all the $n^{O(k)}$ possible edge sets of size $k$ and check whether the addition of these edges makes the graph chordal. This trivial $n^{O(k)}$ time algorithm can be improved to $O(4^k/(k+1)^{3/2} \cdot (n+m))$ time [3] or $O(k^2nm + k^6 2^{4k})$ time [16]. Therefore, chordal edge completion (which is also called the *minimum fill-in problem*) is fixed-parameter tractable. The main result of the paper is that chordal vertex deletion and chordal edge deletion are also fixed-parameter tractable. In fact, we give an algorithm for the common generalization of the two deletion problems: in the CHORDAL DELETION problem the graph has to be made chordal by the deletion of at most $k_1$ vertices and at most $k_2$ edges.

**Theorem 1** CHORDAL DELETION *is fixed-parameter tractable with combined parameters $k_1$ and $k_2$, where $k_1$ (resp., $k_2$) is the maximum number of vertices (resp., edges) to be deleted.*

Cai [4] proposed a general class of graph modification problems analogous to CHORDAL DELETION. Let $\mathcal{G}$ be an arbitrary class of graphs. We denote by $\mathcal{G} + ke$ (resp., $\mathcal{G} - ke$) the class of those graphs that can be obtained by adding (resp., deleting) $k$ edges to/from a member of $\mathcal{G}$. Similarly, let $\mathcal{G} + kv$ contain those graphs that can be obtained from some member of $\mathcal{G}$ by adding $k$ new vertices and connecting these vertices with the original vertices and with each other in an arbitrary way. (An equivalent definition is to say that a graph is in $\mathcal{G} + kv$ if it can be made a member of $\mathcal{G}$ by deleting $k$ vertices.) For every graph class $\mathcal{G}$, we can ask about the complexity of recognizing graphs in $\mathcal{G} + ke$, $\mathcal{G} - ke$, or $\mathcal{G} + kv$. In particular, we are interested in whether these problems are fixed-parameter tractable parameterized by $k$. Our main result implies that recognizing chordal $+ ke$ and chordal $+ kv$ graphs are fixed-parameter tractable. This answers an open question of Cai [4]. The only previous result for this problem is a linear-time algorithm [15] for recognizing chordal $+ 1e$ and chordal $+ 1v$ graphs, which is more efficient than deleting each edge (vertex) and checking whether the remaining graph is chordal.

Our algorithm can actually find the $k$ edges or $k$ vertices whose deletion makes the graph chordal; these edges/vertices are called the *modulator* of the graph in [4]. Vertex coloring of chordal $+ ke$ graphs is fixed-parameter tractable parameterized by $k$, provided that the modulator of the graph is given in the input [21]. The result in this paper implies that the modulator of a chordal $+\, ke$ graph can be generated in $f(k)n^{\alpha}$ time, hence the vertex coloring on chordal $+\, ke$ graphs remains fixed-parameter tractable even if the modulator is not given in the input.

The *iterative compression* method introduced in [24] allows us to concentrate on an easier "solution compression" problem. This technique proved useful for many other problems, see [6, 7, 13]. The compression problem is the following (for brevity, we discuss only the vertex-deletion version in this paragraph): given a set $X$ of $k+1$ vertices such that $G \setminus X$ is chordal, find $k$ vertices whose deletion makes $G$ chordal. To solve this solution compression problem, we first determine the size of the maximum clique in the chordal graph $G \setminus X$. If the clique size $G \setminus X$ is small, then $G \setminus X$ (and hence the slightly larger $G$) has small treewidth. Using standard techniques, the problem can be solved in linear time for graphs with bounded treewidth. On the other hand, we show that if there is a large clique in $G \setminus X$, then the clique contains "irrelevant" vertices that can be removed from the graph without changing the solvability of the problem. The main technical difficulty of the proof is to prove that an irrelevant vertex always exists in a large clique. This idea of repeatedly deleting irrelevant vertices until a bounded-treewidth instance is obtained was useful for other problems as well [12, 22, 25].

The paper is organized as follows. Section 2 reviews some basic facts on chordal graphs. Section 3 presents the algorithm for bounded-treewidth graphs. In Sect. 4 we show how the iterative compression method of [24] can be applied to our problem. Section 5 discusses how we can reduce the size of the cliques to make our graph a bounded treewidth graph.

## 2 Chordal Graphs

We recall some standard definitions from graph theory. A *walk* in a graph $G$ is a sequence of vertices $v_1 v_2 \cdots v_k$ such that $v_i$ and $v_{i+1}$ are adjacent in $G$ for every $1 \le i < k$. The *length* of a walk $v_1 v_2 \cdots v_k$ is defined to be $k-1$. A *path* is walk where the $v_i$'s are distinct. We say that the path $v_1 v_2 \cdots v_k$ *connects* vertices $v_1$ and $v_k$. The *distance* of two vertices $u$ and $v$ is the length of the shortest path connecting $u$ and $v$; the distance is defined to be infinity if there is no such path. The distance of a vertex $v$ and a set $S$ of vertices is the minimum distance of $v$ and a vertex $u \in S$. Vertex $v$ is *adjacent* to $S$ if the distance of $v$ and $S$ is 1, i.e., there is an edge between $v$ and some vertex $u \in S$. A *cycle* in $G$ is a walk $v_1 v_2 \cdots v_k v_{k+1}$ such that $v_1 = v_{k+1}$ and $v_i \ne v_j$ for every $1 \le i < j \le k$. The length of a cycle $v_1 v_2 \cdots v_k v_{k+1}$ is the number of distinct vertices in the sequence, i.e., $k$.

A graph is *chordal* if it does not contain a cycle of length greater than 3 as an induced subgraph. This is equivalent to saying that every cycle of length greater than 3 contains at least one chord, i.e., an edge connecting two vertices not adjacent in the cycle. A chordless cycle of length greater than 3 will be called a *hole*. Chordality is a hereditary property: every induced subgraph of a chordal graph is chordal.

Every chordal graph is a perfect graph [11]: the minimum number of colors required to color the vertices of a chordal graph equals the size of the largest clique.

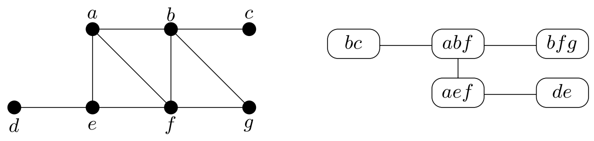

**Fig. 1** A chordal graph and its clique tree decomposition

The complement of a chordal graph is also perfect, which translates to the statement that the minimum number of cliques required to cover the vertices of a chordal graph equals the size of the largest independent set. Furthermore, an optimum coloring or clique covering of a chordal graph can be found in polynomial time [11]. We will use these observations to cover certain sets of vertices with a small number of cliques and treat the cliques separately.

Chordal graphs can be also characterized as the intersection graphs of subtrees of a tree (see e.g., [11]):

**Theorem 2** *The following two statements are equivalent*:

1. *$G(V,E)$ is chordal.*
2. *There exists a tree $T(U,F)$ and a subtree $T_v \subseteq T$ for each $v \in V$ such that $u,v \in V$ are neighbors in $G(V,E)$ if and only if $T_u \cap T_v \neq \emptyset$ (i.e., $T_u$ and $T_v$ have a common node).*

The tree $T$ together with the subtrees $T_v$ is called the *clique tree decomposition* of $G$. Figure 1 shows a chordal graph and a possible clique tree decomposition. The vertices in a node of the tree show which subtrees contain that particular node; for example, the leftmost node of the tree is contained in subtrees $T_b$ and $T_c$. One can find a clique tree decomposition of a given chordal graph in polynomial time (see [11, 26]). For clarity, we will use the word "vertex" when we refer to the graph $G(V,E)$, and "node" when referring to $T(U,F)$. We say that a vertex $v$ *covers* node $x$ if $T_v$ contains node $x$. For an arbitrary node $x$ of $T$, the vertices covering $x$ induce a clique. Conversely, for every clique $K$, there is a node $x$ of $T$ such that every $v \in K$ covers this node $x$ (cf. [11]). The following easy observation will be used repeatedly:

**Proposition 3** *Let $x, y, z$ be vertices in $G(V,E)$ such that $xy, xz \in E$ but $yz \notin E$. If there is a walk $T$ in $G \setminus x$ from $y$ to $z$ such that $y$ and $z$ are the only neighbors of $x$ in $T$, then $T \cup x$ contains a hole of length at least* 4.

*Proof* Let $P$ be a minimal subpath of $T$ from $y$ to $z$. Since $y$ and $z$ are not neighbors, path $P$ has length at least 2. Therefore, the length of $xyPzx$ is at least 4, and it is chordless, since $P$ is a minimal path and $x$ is not the neighbor of the internal vertices of $P$. □

Proposition 3 can be also thought of as a characterization of chordal graphs: if $v_1 v_2 \cdots v_t v_1$ is a hole, then choosing $x = v_1$, $y = v_2$, $z = v_t$ satisfies the requirements.

If the deletion of $X \subseteq V$ and $Y \subseteq E$ makes the graph $G(V,E)$ chordal, then we say that the pair $(X,Y)$ is a *hole cover* of $G$. We use the notation $G \setminus (X,Y)$ for the graph obtained by deleting the vertices $X$ and the edges $Y$ from $G$. The *size* of a hole cover $(X, Y)$ is the pair $(|X|, |Y|)$. We say that a hole cover $(X, Y)$ *obstructs* a path $P$ if $X$ contains a vertex of $P$ or $Y$ contains an edge of $P$. For a hole cover $(X, Y)$, let $\omega(X, Y)$ contain the vertices of $V$ and the endpoints of the edges in $E$; clearly $|\omega(X, Y)| \leq |X| + 2|Y|$.

The problem studied in this paper is formally defined as follows:

CHORDAL DELETION

*Input:* A graph $G(V, E)$ and integers $k_1, k_2$

*Parameter:* $k_1, k_2$

*Task:* Find a hole cover of size $(k_1, k_2)$.

It turns out that the deletion problem is very different from the edge completion problem. The algorithms in [3, 16] for chordal edge completion use the standard method of bounded search trees. If there is a chordless cycle of length more than $k+3$, then the answer is no, as we would need more than $k$ edges to make this cycle chordal. If there is a chordless cycle of length $\ell \leq k+3$, then every solution has to contain $\ell - 3$ edges that make this chordless cycle chordal. There is a constant number of different ways of making a hole of size $\ell$ chordal using $\ell - 3$ edges. The algorithm tries all these possibilities: we branch off into at most a constant (i.e., depending only on $k$) number of directions. After making the cycle chordal, the problem parameter (the number of edges that can be added) is decreased by $\ell - 3$, and the algorithm continues with the next chordless cycle. Since the problem parameter can be decreased only at most $k$ times, the algorithm finishes after at most $k$ branchings. At each step, the number of directions we branch into can be bounded by a function of $k$, thus the size of the search space explored by the algorithm can be also bounded by a function of $k$. In summary, the main idea is that the graph cannot contain a large hole, otherwise the graph could not be made chordal by adding $k$ edges. In the deletion problem we cannot make this assumption: it is possible that the graph can be made chordal by deleting few vertices, even if there are large holes (for example, if the graph is a large chordless cycle, then it can be made chordal by the deletion of a single vertex). This means that there might be many possibilities to repair a long chordless cycle, thus we cannot use the bounded search tree method. Substantially different (and more complicated) ideas are required for the vertex deletion problem.

## 3 Bounded-Treewidth Graphs

One way to define *treewidth* is the following: the treewidth of a graph $G$ is the smallest integer $k$ such that $G$ is a subgraph of a chordal graph $H$ having clique number $k + 1$. Graphs with treewidth 1 are exactly the forests. For more background on treewidth, see for example [2, 18].

The algorithmic importance of treewidth comes from the fact that a large number of NP-hard problems can be solved in linear time if we have a bound on the treewidth of the input graph. Most of these algorithms use a bottom-up dynamic programming approach, which generalizes dynamic programming on trees.

Courcelle's Theorem [5] (see also [8, Sect. 6.5]) gives a powerful way of quickly showing that a problem is linear-time solvable on bounded treewidth graphs. Sentences in the *Extended Monadic Second Order Logic of Graphs* (EMSO) contain quantifiers, logical connectives ($\neg$, $\vee$, and $\wedge$), vertex variables, edge variables, vertex set variables, edge set variables, and the following binary relations: $\in$, $=$, $\mathrm{inc}(e, v)$ (edge variable $e$ is incident to vertex variable $v$), and $\mathrm{adj}(u, v)$ (vertex variables $u$, $v$ are neighbors). If a graph property can be described in this language, then this description can be turned into an algorithm:

**Theorem 4** (Courcelle [5]) *If a graph property can be described in the Extended Monadic Second Order Logic of Graphs, then for every $w$, there is a linear-time algorithm for the recognition of this property on graphs with treewidth at most $w$.*

Using Proposition 3, it is not difficult to describe those graphs $G(V, E)$ that can be made chordal by the deletion of at most $k_1$ vertices and at most $k_2$ edges:

$$\begin{aligned}
&(k_1, k_2)\text{-chordal-deletion}(V, E)\\
&\quad := \exists v_1, \dots v_{k_1} \in V, \exists e_1, \dots, e_{k_2} \in E, V_0 \subseteq V, E_0 \subseteq E :\\
&\qquad \big[\mathrm{chordal}(V_0, E_0) \ \wedge\ (\forall v \in V : v \in V_0 \vee v = v_1 \vee \cdots \vee v = v_{k_1})\\
&\qquad \wedge (\forall e \in E : e \in E_0 \vee e = e_1 \vee \cdots \vee e = e_{k_2})\big],\\
&\mathrm{chordal}(V_0, E_0)\\
&\quad := \neg(\exists x, y, z \in V_0, V_1 \subseteq V_0, E_1 \subseteq E_0 : \mathrm{adj}(x, y) \wedge \mathrm{adj}(x, z) \wedge \neg\mathrm{adj}(y, z)\\
&\qquad \wedge (\forall q \in V_1 : q = y \vee q = z \vee \neg\mathrm{adj}(q, x)) \wedge \mathrm{connected}(y, z, V_1, E_1)),\\
&\mathrm{connected}(y, z, V, E)\\
&\quad := \forall Y, Z \subseteq V : [(\mathrm{partition}(V, Y, Z) \wedge y \in Y \wedge z \in Z)\\
&\qquad \to (\exists y' \in Y, z' \in Z, e \in E : \mathrm{inc}(e, y') \wedge \mathrm{inc}(e, z'))],\\
&\mathrm{partition}(V, Y, Z) := \forall v \in V : (v \in Y \vee v \in Z) \wedge (v \notin Y \vee v \notin Z).
\end{aligned}$$

The predicate $\mathrm{chordal}(V_0, E_0)$ expresses that the subgraph with vertex set $V_0$ and edge set $E_0$ is a chordal graph. To test whether the subgraph is chordal, we check whether there are vertices $x$, $y$, and $z$ satisfying the requirements of Proposition 3, i.e., there is at path $P$ with vertices $V_1$ and edges $E_1$ that connect $y$ and $z$ in such a way that the internal vertices are not adjacent to $x$. To ensure that $y$ and $z$ are connected by the path $P$, we require that for every partition $Y$, $Z$ of $V_1$, if $y \in Y$ and $z \in Z$, then there is an edge of $P$ connecting $Y$ and $Z$.

Courcelle's Theorem together with the EMSO formulation of CHORDAL DELETION implies:

**Theorem 5** *For every $k_1$, $k_2$, and $w$,* CHORDAL DELETION *can be solved in linear time for graphs with treewidth at most $w$.*

We note that Theorem 5 can be obtained without Courcelle's Theorem using standard (but very tedious and technical) dynamic programming techniques.

## 4 Iterative Compression

Reed, Smith and Vetta [24] have shown that the Bipartite Vertex Deletion problem (make the graph bipartite by deleting $k$ vertices) is fixed-parameter tractable. They introduced the method of *iterative compression*, which can be used in the case of the Chordal Deletion problem as well. The idea is that it is sufficient to show that the following easier problem is fixed-parameter tractable:

Hole Cover Compression

*Input:* A graph $G$, integers $k_1, k_2$, and a hole cover $(X, Y)$ of size $(k_1 + 1, k_2)$.

*Parameter:* $k_1, k_2$

*Task:* Find a hole cover $(X', Y')$ of size $(k_1, k_2)$ in $G$.

This problem is easier than Chordal Deletion: the extra input $(X, Y)$ gives us useful structural information about $G$. In particular, we know that $G \setminus (X, Y)$ is chordal. Our algorithm builds heavily on this fact.

Assume that we have an algorithm with running time $f(k_1, k_2)n^\alpha$ for Hole Cover Compression, then Chordal Deletion can be solved as follows (see Fig. 2). Let $v_1, v_2, \dots, v_n$ be an ordering of the vertices, and let $G_i$ be the graph induced by $v_1, \dots, v_i$. We try to find a size-$(k_1, k_2)$ hole cover for each $G_i$. Graph $G_{k_1}$ trivially has such a hole cover. Now assume that $G_i$ has a size-$(k_1, k_2)$ hole cover $(X, Y)$. Clearly, $(X \cup v_{i+1}, Y)$ is a size-$(k_1 + 1, k_2)$ hole cover of $G_{i+1}$. Therefore, the compression algorithm can be used to find a size-$(k_1, k_2)$ hole cover for $G_{i+1}$. If there is such a hole cover, then we can proceed to $G_{i+2}$. Otherwise the answer is no, we can conclude that the supergraph $G$ of $G_{i+1}$ cannot have a size-$(k_1, k_2)$ hole cover either. The algorithm calls the compression method at most $n$ times, thus the total running time is $f(k_1, k_2)n^{\alpha+1}$, which shows that the problem is fixed-parameter tractable. Note that $G_{i+1}$ is obtained from $G_i$ by adding a new vertex (rather than an edge), thus the compression algorithm is invoked with parameter $(k_1 + 1, k_2)$ and not with $(k_1, k_2 + 1)$.

Now let us turn our attention to the Hole Cover Compression algorithm itself. Assume that a size-$(k_1 + 1, k_2)$ hole cover $(X, Y)$ of $G$ is given. Let $W := \omega(X, Y)$, let $V_0 = V \setminus W$, and denote by $G_0$ the chordal graph $G \setminus W$. If the size of the maximum clique in $V_0$ is $c$, then the treewidth of the chordal graph $G_0$ is $c - 1$, and the

```
Chordal Deletion($G, k_1, k_2$)

1. Set $i := k_1$ and let $X$ be the vertices of $G_{k_1}$ and $Y$ be $k_2$ arbitrary edges.
2. Invariant condition: $(X, Y)$ is a size-$(k_1, k_2)$ hole cover of $G_i$.
3. If $i = n$, then return "$(X, Y)$ is a size-$(k_1, k_2)$ hole cover of $G$."
4. Set $X := X \cup v_{i+1}$, now $(X, Y)$ is a size-$(k_1 + 1, k_2)$ hole cover of $G_{i+1}$.
5. Call Hole Cover Compression($G_{i+1}, k_1, k_2, X, Y$).
   – If the answer is a size-$(k_1, k_2)$ hole cover $(X', Y')$ of $G_{i+1}$, then let
     $(X, Y) := (X', Y')$, $i := i + 1$, and go to Step 2.
   – If the answer is "no," then return "no."
```

**Fig. 2** Algorithm Chordal Deletion

treewidth of $G$ is at most $c-1+|W| \leq c-1+k_1+2k_2+1$. Therefore, if the clique size of $G_0$ can be bounded by a constant depending on $k_1$ and $k_2$, then the method described for bounded-treewidth graphs in Sect. 3 can be used to decide whether $G$ has a size-$(k_1, k_2)$ hole cover.

In Sect. 5, we present a method of reducing the clique size of $G_0$ to a constant depending only on $k_1, k_2$. A vertex $v \in V$ is *irrelevant* if every size-$(k_1, k_2)$ hole cover of $G \setminus v$ is also a hole cover of $G$. If we identify an irrelevant vertex $v$, then the problem can be reduced to finding a size-$(k_1, k_2)$ hole cover in $G \setminus v$. We show that if there is a clique $K$ in $G_0$ whose size is greater than some constant $c_{k_1,k_2}$, then the problem can be reduced to a simpler form: either we find an irrelevant vertex or a small set of vertices/edges such that every size-$(k_1, k_2)$ hole cover contains at least one member of this set. More precisely, for a set $N_v$ of vertices and set $N_e$ of edges we say that $(N_v, N_e)$ is a *necessary set* if whenever $(X, Y)$ is a size-$(k_1, k_2)$ hole cover, then either $X$ contains a vertex of $N_v$ or $Y$ contains an edge of $N_e$. If the set $(N_v, N_e) = (\emptyset, \emptyset)$ is a necessary set, then this means that there is no hole cover of the required size. The necessary sets that we find are always small, i.e., there is a constant $b_{k_1,k_2}$ such that $|N_v| + |N_e| \leq b_{k_1,k_2}$. (In the following, when we say “a necessary set can be found,” we always mean that the size of this set can be bounded by a function of $k_1$ and $k_2$.)

If the clique reduction algorithm returns a necessary set $(N_v, N_e)$, then we can conclude that every size-$(k_1, k_2)$ hole cover contains at least one vertex of $N_v$ or an edge of $N_e$. Therefore, we branch into $|N_v| + |N_e|$ directions: for each vertex $v$ of $N_v$, we check whether there is a size-$(k_1 - 1, k_2)$ hole cover of $G \setminus v$ and for each edge of $N_e$, we check whether there is a size-$(k_1, k_2 - 1)$ hole cover. Thus the problem can be reduced to at most $b_{k_1,k_2}$ subproblems with smaller parameter values, where $b_{k_1,k_2}$ depends only on $k$.

In summary, the clique reduction algorithm does one of the following:

- *Identifies an irrelevant vertex* $v \in K$. In this case, the deletion of $v$ does not change the problem. If the maximum clique size is still larger than $c_{k_1,k_2}$, then the algorithm can be applied again. Otherwise, we can use the algorithm of Theorem 5.
- *Identifies a necessary set* $(N_v, N_e)$ *whose size is bounded by a function of* $k_1$ *and* $k_2$. In this case, the algorithm can branch into a constant number of directions: one vertex of $N_v$ or one edge of $N_e$ has to be deleted.

The overall algorithm Hole Cover Compression is shown in Fig. 3. The algorithm calls the clique reduction method (which is described in the following section) and can make some number of recursive calls to Hole Cover Compression with parameter $(k_1 - 1, k_2)$ and with parameter $(k_1, k_2 - 1)$. That is, the sum $k_1 + k_2$ strictly decreases in each recursive call, hence the recursion depth is at most $k_1 + k_2$. By assumption, if Clique Reduction returns a necessary set, then its size can be bounded by a function of $k_1$ and $k_2$. This means that the algorithm branches into a constant number of directions, and the size of the recursion tree can be also bounded by some function of $k_1$ and $k_2$. Thus the running time of Hole Cover Compression can be bounded by $g(k_1, k_2)n^{\alpha}$ for an appropriate function $g$ and constant $\alpha$.

```
HOLE COVER COMPRESSION($G, k_1, k_2, X, Y$)

1. Let $W := \omega(X, Y)$. If the clique size of $G \setminus W$ is at most $c_{k_1,k_2}$, then use the
   algorithm of Theorem 5.
2. If $G \setminus W$ has a clique $K$ of size more than $c_{k_1,k_2}$, then call CLIQUE REDUC-
   TION($G, W, K, k_1, k_2$).
3. If there is an irrelevant vertex $v$, then delete $v$ from $G$, and go to Step 1.
4. If there is a necessary set $(N_v, N_e)$:
5. For each vertex $v \in N_v$, call HOLE COVER COMPRESSION($G \setminus v, k_1 - 1, k_2$).
   – If the answer is "Yes" for some $v \in N_v$, and $(X', Y')$ is a size-$(k_1 - 1, k_2)$ hole
     cover of $G \setminus v$, then answer "$(X' \cup v, Y')$ is a size-$(k_1, k_2)$ hole cover of $G$."
6. For each edge $e \in N_e$, call HOLE COVER COMPRESSION($G \setminus e, k_1, k_2 - 1$).
   – If the answer is "Yes" for some $e \in N_e$, and $(X', Y')$ is a size-$(k_1, k_2 - 1)$ hole
     cover of $G \setminus e$, then answer "$(X', Y' \cup e)$ is a size-$(k_1, k_2)$ hole cover of $G$."
7. If the answer is "No" for every $v$ and every $e$, then answer "No."
```

**Fig. 3** Algorithm HOLE COVER COMPRESSION

## 5 Clique reduction

As in the previous section, we assume that $W$ is a set of at most $k_1 + 2k_2 + 1$ vertices such that $G_0 := G \setminus W$ is a chordal graph. In this section we show that if there is a large clique $K$ in $G_0$, then in polynomial time we can either find a necessary set or an irrelevant vertex of $K$. In the rest of the section, we fix a clique $K$ in $G_0$. Intuitively speaking, a vertex $v$ of $K$ is not irrelevant, if it is somehow essential for the holes of $G$. Every hole of $G$ goes through a vertex of $W$, thus every hole of $G$ not completely contained in $W$ goes through a neighbor of $W$ in $G_0$. Thus the neighbors of $W$ play an important role, hence we try to understand the structure of such vertices in Sect. 5.1. Those neighbors of $W$ are especially important that are reachable from $K$ in certain technical sense, and hence can be part of a hole containing also a vertex of $K$. We will investigate such vertices in Sect. 5.2. These structural results enable us to identify a bounded number of important vertices in the clique $K$ and we can declare any other vertex of the clique irrelevant (Sect. 5.3). More precisely, in Sect. 5.4 we show that if there is a hole going through such an irrelevant vertex (possibly after the deletion of $k_1$ vertices and $k_2$ edges), then there is a hole avoiding this vertex. This shows that removing the irrelevant vertex does not change the answer to the problem.

### 5.1 Labeling

If a vertex $v \in V \setminus W$ is the neighbor of some vertex $\ell \in W$, then we say that $v$ *has label* $\ell$. A vertex can have more than one label; the labels of a given vertex form a subset of $W$. The following easy observations will be used to find necessary sets if certain structures appear in the graph $G_0$:

**Proposition 6** *If $P$ is a path of length at least 2 connecting nonadjacent vertices $u$ and $v$, and vertices $u$ and $v$ are the only vertices in $P$ having label $\ell$, then every hole cover has to contain either $\ell$, $\ell u$, $\ell v$ or at least one vertex or edge of $P$.*

*Proof* If $(X, Y)$ is a hole cover disjoint from $P$ and contains none of vertex $\ell$, edges $\ell u$, and $\ell v$, then $\ell u P v \ell$ contains a hole in $G \setminus (X, Y)$ (Proposition 3), a contradiction. □

**Lemma 7** *Let $v$ be a vertex without label $t$, let $x_1, \ldots, x_{k_1+k_2+2}$ be independent $t$-labeled vertices, and let $P_1, \ldots, P_{k_1+k_2+2}$ be internally disjoint paths where $P_i$ connects $v$ and $x_i$, and the internal vertices of $P_i$ do not have label $t$. Then $(\{v, t\}, \emptyset)$ is a necessary set.*

*Proof* Let $(X, Y)$ be a hole cover of size-$(k_1, k_2)$ disjoint from $(\{v, t\}, \emptyset)$. Consider the internally disjoint paths $v P_i x_i t$ for every $i = 1, \ldots, k_1 + k_2 + 2$. Since $v, t \notin X$, hole cover $(X, Y)$ can obstruct at most $k_1 + k_2$ of these paths. Assume without loss of generality that $v P_1 x_1 t$ and $v P_2 x_2 t$ are not obstructed; this means that $x_1$ and $x_2$ can be connected with a path $x_1 P_1 v P_2 x_2$ whose internal vertices do not have label $t$. Since $x_1$ and $x_2$ are neighbors of $t$ in $G \setminus (X, Y)$ and there is no edge between them, Proposition 6 implies that there is a hole in $G \setminus (X, Y)$. □

**Lemma 8** *Let $H_1, \ldots, H_{k_1+k_2+1}$ be holes in $G$, let $S$ be the set of all vertices that are contained in more than one $H_i$, and let $E_S$ be the edges induced by $S$. If $|S| \le c$ for some constant $c$ depending only on $k_1$ and $k_2$, then $(S, E_S)$ is a necessary set of size at most $c + c(c-1)/2$.*

*Proof* Let $(X, Y)$ be a hole cover of size-$(k_1, k_2)$ such that $S \cap X = \emptyset$ and $S_E \cap Y = \emptyset$. Now each vertex of $X$ and each edge of $Y$ can be contained in at most one hole $H_i$. Thus there has to be a hole which is not covered by $(X, Y)$, a contradiction. □

In Lemma 10 we give a bound on the number of independent labeled vertices in the neighborhood of a connected unlabeled set. We need the following lemma of Kleinberg [17]:

**Lemma 9** (Kleinberg [17]) *Let $A$ be a set of vertices. Suppose that for some $k$, there do not exists $k + 1$ pairwise disjoint paths with distinct endpoints in $A$. Then there is a set $Z$ of size at most $3k$ such that each component of $G \setminus Z$ contains at most one vertex of $A \setminus Z$.*

Note that there is a polynomial-time algorithm that finds $k + 1$ pairwise disjoint paths with distinct endpoints in $A$ (if such paths exist) [10] and the proof of Lemma 9 can be made algorithmic. Thus in polynomial time we can either find the $k+1$ disjoint paths or the set $Z$ of size $3k$.

**Lemma 10** *Let $B$ be a connected subset of $V_0 = V(G_0)$ such that no vertex in $B$ has label $t$. Let $I$ be an independent set of $t$-labeled vertices in the neighborhood of $B$. If $|I| > 6(k_1 + k_2)^2$, then we can find a necessary set in polynomial time.*

*Proof* Let $I = \{v_1, v_2, \ldots, v_{6(k_1+k_2)^2+1}\}$ be an independent set of vertices with label $t$ in the neighborhood of $B$. Denote by $G_0'$ the subgraph of $G_0$ induced by $I \cup B$. If there are $k_1 + k_2 + 1$ disjoint paths in $G_0'$ with distinct endpoints in $I$, then these paths together with vertex $t$ give $k_1 + k_2 + 1$ holes that intersect only in vertex $t$. By Lemma 8, this means that we can find a necessary set. Assume therefore that there are no such paths; by Lemma 9, this means that there is a set $Z$ of size at most $3k_1 + 3k_2$ such that each component of $G'_0 \setminus Z$ contains at most one vertex of $I$. Let $C_1, \dots, C_c$ be the components of $G'_0 \setminus Z$ containing a vertex of $I$, and let $v_i$ be the unique vertex $C_i \cap I$. Note that $c \geq |I \setminus Z| \geq 6(k_1 + k_2)^2 + 1 - 3(k_1 + k_2) > 3(k_1 + k_2)(k_1 + k_2 + 1)$ (if $k_1 + k_2 > 1$).

We claim that each $C_i$ is adjacent to a vertex of $Z \cap B$. First, it is not possible that $Z \cap B = \emptyset$: vertices $v_i$ and $v_j$ are in the neighborhood of $B$, hence they can be connected with a path whose internal vertices are in $B$, and this path would not be blocked by $Z$ if $Z \cap B = \emptyset$. Let $z \in Z \cap B$ be an arbitrary vertex. Each vertex $v_i$ has a neighbor $u \in B$. If $u \in Z$, then $u$ is a neighbor of $C_i$ in $Z \cap B$. Otherwise, there is a path fully contained in $B$ that connects $u$ and $z$. Let $z'$ be the first vertex (starting from $u$) on this path that is in $Z$. Now $z'$ is a neighbor of $C_i$.

Since $|Z \cap B| \leq 3(k_1 + k_2)$, there has to be a vertex $z \in Z \cap B$ that is adjacent to more than $k_1 + k_2 + 1$ components. Assume without loss of generality that $z$ is adjacent to components $C_1, \dots, C_{k_1+k_2+2}$, and path $P_i$ connects vertex $v_i$ with $z$ such that the internal vertices of $P_i$ are in $C_i$. Note that these paths intersect only in $Z \cap B$. Since $z \in Z \cap B$ does not have label $t$, Lemma 7 gives a necessary set. □

## 5.2 Dangerous Vertices

Let us fix a maximal clique $K$ of $G_0$. A vertex $v \in V_0 \setminus K$ is called a *$t$-dangerous vertex* (for $K$) if $v$ has label $t$ and there is a path $P$ from $v$ to a vertex $u \in K$ such that $v$ is the only vertex having label $t$ on the path. Vertex $v$ is a *$t^*$-dangerous vertex* if $v$ has label $t$ and there is a path $P$ from $v$ to a vertex $u \in K$ such that $v$ and $u$ are not neighbors, $u$ also has label $t$, and the internal vertices of the path do not have label $t$. Vertex $u$ is a *$t$-witness* (*$t^*$-witness*) of $v$, the path $P$ is a *$t$-witness* (*$t^*$-witness*) *path* of $v$. A vertex $v$ can be $t$-dangerous for more than one $t \in W$, or it can be $t$- and $t^*$-dangerous at the same time. For a subgraph $G'_0$ of $G_0$, we use the expression *with respect to $G'_0$* if we require that the witness path is in $G'_0$.

The name dangerous comes from the observation that if there is a hole in $G$ that goes through the clique $K$, then the hole has to go through a dangerous vertex as well. For example, if a hole starts in $t \in W$, goes to a $t$-labeled neighbor $v \in V_0 \setminus K$ of $t$, goes to a $t$-labeled vertex $u \in K$ via a path $P \subseteq V_0$, and returns to $t$, then $v$ is a $t^*$-dangerous vertex, $u$ is its witness, and $P$ is the witness path (see Fig. 4a). In the situation depicted in Fig. 4b, the hole goes through two vertices $t_1, t_2$ of $W$, and the hole has a subpath with endpoints $v_1, v_2$ that goes through $K$ (where $v_1$ and $v_2$ are the neighbors of $t_1$ and $t_2$, respectively). The internal vertices of this path do not have labels $t_1, t_2$, hence $v_1$ is $t_1$-dangerous and $v_2$ is $t_2$-dangerous, and $u$ is a witness for both. When we delete vertices to make the graph chordal, our aim is to destroy as many witness paths as possible and to make many vertices non-dangerous. It will turn out that if a clique is large, then it contains many vertices whose deletion does not affect the dangerous vertices, thus there is no use of deleting them.

We prove two technical results on dangerous vertices: we bound by $6(k_1 + k_2)^2$ (resp., $6(k_1 + k_2)^3$) the number of independent $t$-dangerous (resp., $t^*$-dangerous) ver-

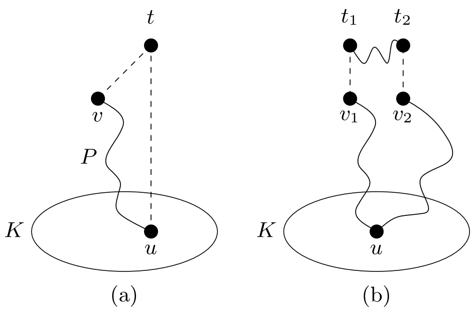

**Fig. 4** (**a**) A $t^*$-dangerous vertex $v$. (**b**) A $t_1$-dangerous vertex $v_1$ and a $t_2$-dangerous vertex $v_2$

tices. Since $G_0$ is chordal (hence perfect), it follows that these vertices can be covered by $6(k_1+k_2)^2$ (resp., $6(k_1+k_2)^3$) cliques.

**Lemma 11** *Given a set $I$ of more than $6(k_1+k_2)^2$ independent $t$-dangerous vertices, we can find a necessary set in polynomial time.*

*Proof* Consider the subgraph $G_0'$ of $G_0$ induced by those vertices that do not have label $t$. The clique $K$ contains vertices only from one connected component of $G_0'$, let $B$ be this component. Clearly, every $t$-dangerous vertex is a neighbor of $B$ in $G_0$. Therefore, by Lemma 10, we can find a necessary set. □

**Lemma 12** *Given a set $I$ of more than $6(k_1+k_2)^3$ independent $t^*$-dangerous vertices, we can find a necessary set in polynomial time.*

*Proof* Consider the subgraph $G_0'$ of $G_0$ induced by the vertices without label $t$. Let $C_1, \ldots, C_c$ be the connected components of $G_0'$. The internal vertices of a witness path for a $t^*$-dangerous vertex are completely contained in one of these components. Let $I_i \subseteq I$ contain a $t^*$-dangerous vertex $v \in I$ if and only if $v$ has a witness path with internal vertices only in $C_i$.

If $|I_i| > 6(k_1+k_2)^2$ for some $1 \le i \le c$, then we are ready by using Lemma 10 for the connected subgraph $C_i$. Thus $c > k_1+k_2$, otherwise the size of the independent set is at most $6(k_1+k_2)^3$. Let us fix $k_1+k_2+1$ of these components. For each such component $C_i$, let us select a $t^*$-dangerous vertex that has a witness path $P_i$ whose internal vertices are in $C_i$. Each path $P_i$ together with vertex $t$ form a hole. As the internal vertices of the $P_i$'s are in different components, the $k_1+k_2+1$ holes can intersect each other only in their endpoints and in $t$. This means that there are only $2k_1+2k_2+3$ vertices that are contained in more than one of the holes; therefore, by Lemma 8, we can find a necessary set of bounded size. □

## 5.3 Marking the Clique

In the next two lemmas, we show that for a clique $Q$ of dangerous vertices, there is only a constant (i.e., depending only on $k_1, k_2$) number of vertices in $K$ whose deletion can make a dangerous vertex of $Q$ non-dangerous. For every other vertex $u \in K$, if $v$ is $t$-dangerous, then $v \in Q$ remains $t$-dangerous with respect to $G_0 \setminus u$.

Even more is true: if $X$ is a set of at most $k_1$ vertices and $Y$ is a set of at most $k_2$ edges, then $v \in Q$ is $t$-dangerous with respect to $G_0 \setminus (X, Y)$ if and only if $v$ is $t$-dangerous with respect to $G_0 \setminus (X \cup u, Y)$. In the following lemma, we mark some number of vertices such that any unmarked vertex $u \in K$ has this property. Essentially, we have to mark those vertices of $K$ that are "closest" to $Q$, where closeness is measured in the clique tree decomposition.

**Lemma 13** *Let $Q$ be a clique of $t$-dangerous vertices. For every $k_1, k_2$, there is a constant $d_{k_1,k_2}$, such that we can mark $d_{k_1,k_2}$ vertices in $K$ such that if $X$ is a set of $k_1$ vertices, and $Y$ is a set of $k_2$ edges, and $v \in Q$ has an unmarked $t$-witness $u$ with respect to $G_0 \setminus (X, Y)$, then $v$ has a marked $t$-witness $u' \in K \setminus \omega(X, Y)$ with respect to $G_0 \setminus (X \cup u, Y)$.*

*Proof* Consider the clique tree decomposition of the chordal graph $G_0$. Since $Q$ and $K$ are cliques, there are two nodes $x$ and $y$ such that every vertex of $Q$ covers node $x$, and every vertex of $K$ covers node $y$. Consider those vertices of $K$ that do not have label $t$, and order these vertices such that the distance of their subtrees from node $x$ is nondecreasing. Let us mark the first $d_{k_1,k_2} := k_1 + 2k_2 + 1$ vertices (or all of them, if there are less than $k_1 + 2k_2 + 1$ such vertices). Suppose that the witness $u$ of $v$ is not marked. Since $|\omega(X, Y)| \leq k_1 + 2k_2$, there is a marked vertex $u' \in K \setminus \omega(X, Y)$. By the way the vertices are ordered, the distance of the subtree of $u'$ from $x$ is not larger than the distance of the subtree of $u$ from $x$. Therefore, the witness path $P$ connecting $v$ and $u$ goes through the neighborhood of $u'$, i.e., $P$ has a subpath $P'$ from $v$ to a neighbor $w$ of $u'$. As $u' \notin \omega(X, Y)$, the edge $wu'$ is in $G_0 \setminus (X, Y)$, hence the witness path $vP'u'$ shows that $u'$ is a $t$-witness of $v$ with respect to $G \setminus (X \cup u, Y)$. □

The next lemma proves a similar statement for $t^*$-dangerous vertices. However, now the marking procedure is more complicated. The reason for this complication is that a $t^*$-witness for $v$ has to satisfy two (somewhat contradicting) requirements: the witness has to be reachable from $v$ (thus it has to be close to the clique $Q$), but it should not be a neighbor of $v$ (thus it should not be too close to $Q$).

**Lemma 14** *Let $Q$ be a clique of $t^*$-dangerous vertices. For every $k_1, k_2$, there is a constant $d^*_{k_1,k_2}$ such that either we can find a necessary set or we can mark $d^*_{k_1,k_2}$ vertices in $K$ such that if $X$ is a set of $k_1$ vertices, $Y$ is a set of $k_2$ edges, $v \in Q$ has an unmarked $t^*$-witness with respect to $G_0 \setminus (X, Y)$, then $v$ has a marked $t^*$-witness $u \in K \setminus \omega(X, Y)$ as well.*

*Proof* Consider the clique tree decomposition of the chordal graph $G_0$, let $T_v$ be the subtree corresponding to a vertex $v$. Since $Q$ and $K$ are cliques, there are two nodes $x$ and $y$ such that every $v \in Q$ covers $x$, and every $u \in K$ covers $y$. Consider the unique path connecting $x$ and $y$ in the tree, and identify the vertices of the path with the integers $1, 2, \ldots, n$, where $x = 1$ and $y = n$. Let $u_1, u_2, \ldots$ be the vertices of $K$ having label $t$ and denote by $a_i$ the smallest node of $T_{u_i}$ on this path. Similarly, let $v_1, v_2, \ldots$ be the vertices of $Q$ and denote by $b_i$ the largest node of $T_{v_i}$ on this path. Clearly, $T_{v_i}$ and $T_{u_j}$ intersect if and only if $a_i \leq b_j$. For convenience, we assume that

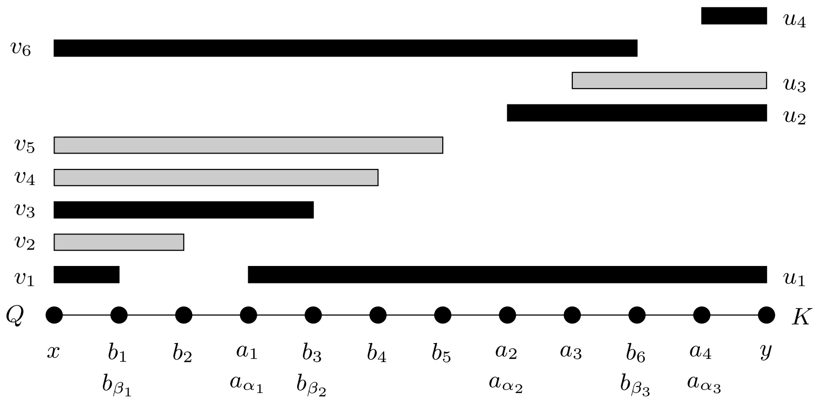
$u_4$
$v_6$
$u_3$
$u_2$
$v_5$
$v_4$
$v_3$
$v_2$
$v_1$
$u_1$
$Q$
$K$
$x$ $b_1$ $b_2$ $a_1$ $b_3$ $b_4$ $b_5$ $a_2$ $a_3$ $b_6$ $a_4$ $y$
$b_{\beta_1}$ $a_{\alpha_1}$ $b_{\beta_2}$ $a_{\alpha_2}$ $b_{\beta_3}$ $a_{\alpha_3}$

**Fig. 5** Proof of Lemma 14: the path between nodes $x$ and $y$. The rectangles show the subtrees of the $v_i$'s and $u_i$'s on this path

the $a_i$'s and $b_i$'s are all distinct, this can be achieved by slightly modifying the tree decomposition. Furthermore, we can assume that the vertices are ordered such that the sequence $a_i$ and the sequence $b_i$ are strictly increasing (see Fig. 5).

We define a subsequence of $b_i$ and $a_j$ as follows. Let $\beta_1 = 1$. For every $j \geq 1$, let $\alpha_j$ be the smallest value such that $a_{\alpha_j} > b_{\beta_j}$. For every $i \geq 2$, let $\beta_i$ be the smallest value such that $b_{\beta_i} > a_{\alpha_{i-1}}$. If we cannot find such a $\beta_i$ or $\alpha_j$, then we stop. Therefore, the sequence $b_{\beta_1}, a_{\alpha_1}, b_{\beta_2}, a_{\alpha_2}, \ldots$ is strictly increasing. In Fig. 5, dark rectangles correspond to the members of this sequence.

Let $u_s$ be a witness of a $t^*$-dangerous vertex $v_{\beta_j}$. We claim that $u_{\alpha_j}$ is also a witness for $t^*$-dangerous vertex $v_{\beta_j}$. Clearly, $a_s > b_{\beta_j}$ (otherwise $u_s$ would be a neighbor of $v_{\beta_j}$), hence $a_s \geq a_{\alpha_j}$ by the definition of $\alpha_j$. Let $P$ be a witness path from $v_{\beta_j}$ to $u_s$. Since $a_s \geq a_{\alpha_j}$, path $P$ goes through the neighborhood of $u_{\alpha_j}$, i.e., there is a vertex $w$ of $P$ that is in the neighborhood of $u_{\alpha_j}$. Let $P'$ be the subpath of $P$ from $v_{\beta_j}$ to $w$. As $u_{\alpha_j}$ is not a neighbor of $v_{\beta_j}$ (by construction of the sequence $b_{\beta_1}, a_{\alpha_1}, \ldots$), path $v_{\beta_j} P' u_{\alpha_j}$ is a witness path. This proves the claim that $u_{\alpha_j}$ is a witness of $v_{\beta_j}$.

Let $b_{\beta_\ell}$ be the last element of the sequence that corresponds to a vertex of $Q$. We claim that if $\ell > 2k_1 + 2k_2 + 1$, then we can find a necessary set. Let $P_i$ be a witness path from $v_{\beta_i}$ to its witness $u_{\alpha_i}$. For every $1 \leq i \leq k_1 + k_2 + 1$, let $H_i$ be the hole $t v_{\beta_{2i}} P_{2i} u_{\alpha_{2i}} t$. Suppose first that a vertex $w$ of $G_0$ appears in two holes $H_i$ and $H_{i'}$ for $i < i'$. This is only possible if $w$ is an internal vertex of both $P_{2i}$ and $P_{2i'}$. It is easy to see that each internal vertex of $P_{2i}$ covers at least one node in the interval $[b_{\beta_{2i}}, a_{\alpha_{2i}}]$ and each internal vertex of $P_{2i'}$ covers at least one node in the interval $[b_{\beta_{2i'}}, a_{\alpha_{2i'}}]$. Therefore, $w$ covers both $a_{\alpha_{2i}}$ and $b_{\beta_{2i'}}$ which implies that $w$ also covers $b_{\beta_{2i+1}}$ and $a_{\alpha_{2i+1}}$ (since $2i' > 2i + 1$). Now $t v_{\beta_{2i+1}} w u_{\alpha_{2i+1}}$ is a hole of size 4 and the vertices and edges of this hole form a necessary set. Therefore, we can assume that every vertex of $G_0$ appears in at most one of the holes $H_1, \ldots, H_{k_1+k_2+1}$. Thus there is only one vertex, namely $t$, that appears in more than one of the holes, hence by Lemma 8, $(\{t\}, \emptyset)$ is a necessary set.

Therefore, it can be assumed that $\ell \leq 2k_1 + 2k_2 + 1$. For each $i = 1, 2, \ldots, \ell$, we mark the $k_1 + 2k_2 + 1$ vertices $u_{\alpha_i}, u_{\alpha_i+1}, \ldots, u_{\alpha_i+k_1+2k_2+1}$ (if they exist). Thus we mark at most $d^*_{k_1,k_2} := (k_1+2k_1+1)(2k_1+2k_2+1)$ vertices. Assume that vertex $v_x \in Q$ has a witness path (with respect to $G_0 \setminus (X, Y)$) to some $u_y$. Since $v_x$ and $u_y$ are not neighbors, $b_x < a_y$ and there is a $j$ with $b_x < a_{\alpha_j} \leq a_y$. If $y \leq \alpha_j + k_1 + 2k_2 + 1$, then $u_y$ is marked. Otherwise $\omega(X, Y)$ does not contain at least one of the vertices $u_{\alpha_j+1}, u_{\alpha_j+2}, \ldots, u_{\alpha_j+k_1+2k_2+1}$, say vertex $u_{\alpha_j+r} \notin \omega(X, Y)$. Since $u_y$ is a witness of $v_x$, there is a path $P$ from $v_x$ to $u_y$ in $G \setminus (X, Y)$ such that the internal vertices of $P$ do not have label $t$. From $a_{\alpha_j+r} \leq a_{\alpha_j+k_1+2k_2+1} < a_y$ it follows that $P$ goes through a neighbor $w$ of $a_{\alpha_j+r}$; let $P'$ be the subpath of $P$ from $v_x$ to $w$. Since $u_{\alpha_j+r} \notin \omega(X, Y)$, edge $wu_{\alpha_j+r}$ is in $G \setminus (X, Y)$. Moreover, $b_x < a_{\alpha_j} \leq a_{\alpha_j+r}$ implies that $v_x$ and $u_{\alpha_j+r}$ are not neighbors, thus vertex $u_{\alpha_j+r}$ is a $t^*$-witness of $v_x$ with witness path $v_x P' u_{\alpha_j+r}$. □

In the next two lemmas, we extend Lemmas 13 and 14 to apply not only for a clique $Q$ of $t$-dangerous vertices, but for every dangerous vertex. By Lemmas 11 and 12, there are no large independent sets of dangerous vertices. Observing that $G_0$ is chordal and hence its complement is a perfect graph (as discussed in Sect. 2), we obtain that the number of cliques required to cover the dangerous vertices is a constant depending only on $k_1, k_2$.

**Lemma 15** *For every $k_1, k_2$, there is a constant $c^{(1)}_{k_1,k_2}$ such that either we can find a necessary set or we can mark $c^{(1)}_{k_1,k_2}$ vertices in $K$ such that for every set $X$ of $k_1$ vertices, set $Y$ of $k_2$ edges, and label $t \in W$, if vertex $v$ is a $t$-dangerous vertex $v$ with respect to $G_0 \setminus (X, Y)$ and $v$ has an unmarked witness $u \in K$, then $v$ has a marked witness $u' \in K \setminus \omega(X, Y)$ with respect to $G_0 \setminus (X \cup u, Y)$.*

*Proof* For every $t \in W$, we mark vertices as follows. Consider the set of vertices $D$ that are $t$-dangerous for $K$ in $G_0$. For chordal graphs, a maximum independent set can be found in polynomial time [11]; let $I$ be a maximum independent set in $D$. If $|I| > 6(k_1 + k_2)^2$, then we can find a necessary set by Lemma 11. Thus the size of the maximum independent set in $D$ is at most a constant depending only on $k_1$ and $k_2$. The number of cliques required to cover $D$ is exactly the number of independent sets required to cover $D$ in the complement graph, i.e., it is the chromatic number of the complement of $G[D]$. Since $G[D]$ induces a chordal graph (as $D \subseteq V \setminus W$) and the complement of a chordal graph is a perfect graph [11], it follows that $D$ can be covered by at most $6(k_1 + k_2)^2$ cliques. For each such clique $Q$, we mark the vertices given by Lemma 13. Hence the total number of marked vertices in $K$ can be bounded by a constant depending only on $k_1, k_2$. □

**Lemma 16** *For every $k_1, k_2$, there is a constant $c^{(2)}_{k_1,k_2}$ such that either we can find a necessary set or we can mark $c^{(2)}_{k_1,k_2}$ vertices in $K$ such that for every set $X$ of $k_1$ vertices, set $Y$ of $k_2$ vertices, and label $t \in W$, if a vertex $v$ is $t^*$-dangerous with respect to $G \setminus (X, Y)$ and has an unmarked witness $u \in K$, then $v$ has a marked witness $u \in K \setminus \omega(X, Y)$ with respect to $G_0 \setminus (X \cup u, Y)$ as well.*

*Proof* The proof is similar to the proof of Lemma 15. For each $t \in W$ and each clique $Q$ of $t^*$-dangerous vertices, we mark vertices as in Lemma 14, the rest of the proof is identical. □

## 5.4 Fragments of a Hole

Let $H$ be a hole in $G$. Since $G \setminus W$ is chordal, $H$ has to contain at least one vertex of $W$. Hence $H \setminus W$ is a set of paths $P_1, P_2, \ldots, P_s$, the set $F = H \cap W$ together with this collection of paths will be called the *fragments* of the hole $H$ (Fig. 6). The paths $P_1, \ldots, P_s$ are independent: $P_i$ and $P_j$ do not have adjacent vertices if $i \neq j$. The internal vertices of a path $P_i$ do not have any labels from $F$. Moreover, each end point has exactly one label from $F$. The only exception is that if a path $P_i$ consists of only a single vertex, in this case it contains exactly two labels from $F$ (see $P_1$ in Fig. 6). A label in $F$ can appear only on at most two vertices in the fragments: if a vertex of $W$ is in the hole, then at most two of its neighbors can belong to the hole. However, the neighbors of a vertex in $W$ can also be in $W$, thus it is possible that a label in $F$ appears on only one or on none of the paths. Another property is that if the length of $P_i$ is 1, then the labels of the two end points are different, otherwise the hole would induce a triangle.

The following lemma shows that if we have the fragments of a hole, and a path is replaced with some new path satisfying certain requirements, then the new collection of paths also induces a hole.

**Lemma 17** *Let $F, P_1, \ldots, P_s$ be the fragments of a hole $H$. Assume that the length of $P_1$ is at least 1. Let $x$ and $y$ be the end points of $P_1$, and let $\ell_x$ and $\ell_y$ be their (unique) labels in $F$, respectively. Let $P_1'$ be a path with the following properties:*

- *the end points of $P_1'$ are $x$ and $y'$, for some vertex $y'$ that has label $\ell_y$,*
- *the internal vertices of $P_1'$ do not have label $\ell_x$,*
- *if $\ell_x \neq \ell_y$, then $y'$ does not have label $\ell_x$,*
- *if $\ell_x = \ell_y$, then $x$ and $y'$ are not neighbors.*

*Then there is a hole in the graph induced by the vertices of $F, P_1', P_2, \ldots, P_s$.*

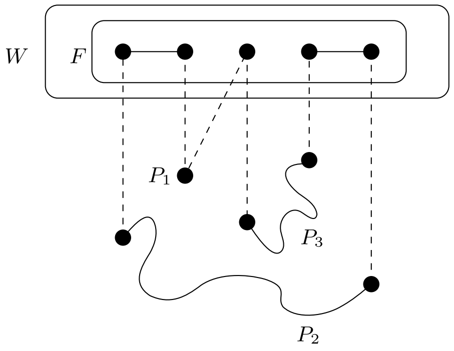

**Fig. 6** The fragments $F$, $P_1$, $P_2$, $P_3$ of a hole

*Proof* We consider two cases. If $|F| = 1$, then $\ell_x = \ell_y$. Since $x$ and $y'$ are not neighbors, the internal vertices of the path $P_1'$ do not have label $\ell_x$, it follows that the path $P_1'$ and the only vertex of $F$ form a hole of length at least 4.

Now assume that $|F| > 1$. It can be assumed that $P_1'$ is a minimal path, i.e., each internal vertex on the path is adjacent only to the previous and the next vertex. Let $z$ be the (unique) neighbor of $x$ on $P_1'$. The paths $P_1'$, $P_2, \dots, P_s$, and the set $F$ gives a walk from $z$ to $\ell_x$ without going through $x$. Furthermore, $z$ and $\ell_x$ are the only vertices on this walk that are in the neighborhood of $x$. To see this, observe that $x$ is adjacent only to $\ell_x$ in $F$, only to $z$ in $P_1'$, and to no vertex in $P_2, \dots, P_s$. As $\ell_x$ and $z$ are not adjacent ($z$ does not have label $\ell_x$), Proposition 3 implies that the graph induced by $F$, $P_1'$, $P_2, \dots, P_s$ contains a hole. □

To show that a vertex $u \in K$ is irrelevant, we have to show that every size-$(k_1, k_2)$ hole cover of $G \setminus u$ is a hole cover of $G$. That is, if $X$ is a set of $k_1$ vertices, $Y$ is a set of $k_2$ edges, and there is a hole $H$ in $G \setminus (X, Y)$ going through $u$, then there is a hole $H'$ in $G \setminus (X \cup u, Y)$. The idea is to look at the fragments of $H$ and reroute one of the paths: if path $P_1$ is going through $u$, then we find a path $P_1'$ avoiding $u$, and use Lemma 17 to obtain the hole $H'$. As we shall see in Lemma 19, if the length of $P_1$ is at least 1, then $P_1'$ can be found using our previous results on dangerous vertices. However, we have to treat separately the case when $P_1$ consists of only a single vertex. This seemingly simple case turns out to be surprisingly difficult.

**Lemma 18** *For every $k_1, k_2$, there is a constant $c^{(3)}_{k_1,k_2}$ such that either we can find a necessary set or we can mark $c^{(3)}_{k_1,k_2}$ vertices in $K$ such that if $X$ is a set of $k_1$ vertices, $Y$ is a set of $k_2$ edges, and there is a hole in $G \setminus (X, Y)$ with fragments $F, P_1, \dots, P_s$ where $P_1$ is only a single vertex $u \in K$, then $G \setminus (X, Y)$ has a hole that does not use any unmarked vertex of $K$.*

*Proof* For every $\ell_1, \ell_2, \ell_3 \in W$, consider those vertices of $K$ that have both labels $\ell_1$ and $\ell_2$, but do not have label $\ell_3$ and let us mark $k_1 + 2k_2 + 1$ of these vertices (if there are less than $k_1 + 2k_2 + 1$ such vertices, then we mark all of them). Since the number of triples $(\ell_1, \ell_2, \ell_3)$ depends only on $|W| \le k_1 + 2k_2 + 1$, the number of marked vertices can be bounded by a function of $k_1, k_2$.

Let $F, P_1, \dots, P_s$ be the fragments of a hole $H$. Without loss of generality, assume that $P_1$ consists of a single vertex $u$, in this case $u$ has two labels $\ell_1, \ell_2$ from $F$. Let us consider the case $|F| > 2$ first. If $|F| > 2$, then there is another label $\ell_3 \in F \setminus \{\ell_1, \ell_2\}$. Vertex $\ell_3$ has two neighbors $a$ and $b$ in the hole $H$, and there is a walk from $a$ to $b$ such that the internal vertices of this walk are not neighbors of $\ell_3$. By the way we marked the vertices, there is a marked vertex $u' \in K \setminus (X, Y)$ that has labels $\ell_1, \ell_2$, but does not have label $\ell_3$. Therefore, if we replace $P_1$ with the path $P_1'$ consisting only of the single vertex $u'$, then we get another walk from $a$ to $b$. Since $u'$ does not have label $\ell_3$, it remains true that the internal vertices of this walk are not neighbors of $\ell_3$. Hence by Proposition 3, there is a walk that contains only the marked vertex $u'$ from $K$.

The hard case is when $|F| = 2$, the rest of the proof is devoted to handle this situation. We mark some additional vertices as follows. If $|F| = 2$, then $s$ cannot be larger than 2. Furthermore, it is not possible that $s = 1$, since that would imply that the hole has only three vertices $\ell_1, \ell_2 \in F$, and $P_1$. Therefore,

(*) hole $H$ has two fragments $P_1$ and $P_2$, where $P_1$ is only a single vertex of $K$.

Consider a clique tree decomposition of $G_0$ and let $x$ be a node that is covered by every vertex of the clique $K$. Assume that $x$ is the root of the tree in the decomposition. For each hole $H$ satisfying (*), define $w_H$ to be the node that is covered by some vertex of $P_2$ and is closest to the node $x$. Observe that $w_H$ cannot be $x$: that would imply that some vertex of $P_2$ is adjacent with every vertex of $K$, including $P_1$. Let $w_1, \dots, w_r$ be those nodes that can arise this way from some hole satisfying (*). Although the number of holes satisfying (*) can be exponential, for every node $w$ we can check in polynomial time whether there is a hole $H$ with $w_H = w$: all we have to do is to try every possible single-vertex path $P_1$ in $K$ and every possible endpoints of $P_2$, and for each possibility check whether there is a suitable path that covers only $w$ and some of its descendants. Assume that the nodes $w_i$ are ordered by nonincreasing distance from $x$. We select a subset of these nodes the following way: we go through the list $w_1, \dots, w_r$, and a select a node if and only if none of its descendants are selected. Let $w_{i_1}, \dots, w_{i_q}$ be the selected nodes. Observe that a selected node cannot be the ancestor or descendant of some other selected node.

We consider two cases. First we show that if $q > k_1 + k_2$, then a necessary set can be identified. Consider the holes $H_{i_1}, \dots, H_{i_{k_1+k_2+1}}$ that give rise to the nodes $w_{i_1}, \dots, w_{i_{k_1+k_2+1}}$. For each hole $H_{i_j}$, there is a path $P_2$ in the fragments of the hole, denote by $P_{i_j}$ this path. By the definition of $w_{i_j}$, the vertices of $P_{i_j}$ cover only the descendants of $w_{i_j}$, hence in particular they do not cover a descendant of $w_{i_{j'}}$ for any $j \neq j'$. It follows that there are at most $k_1 + k_2 + 3$ vertices that appear in more than one of these holes: the vertices $\ell_1, \ell_2$ and at most $k_1 + k_2 + 1$ vertices in $K$. Thus by Lemma 8, we can find a necessary set.

Assume therefore that $q \leq k_1 + k_2$. For each $w_{i_j}$, we mark at most $k_1 + 2k_2 + 1$ vertices of $K$. Consider those vertices of $K$ that have both labels $\ell_1$ and $\ell_2$. For every such vertex $v$, the tree corresponding to $v$ has some distance from node $w_{i_j}$. Order the vertices such that this distance is nonincreasing and mark the first $k_1 + 2k_2 + 1$ vertices in this ordering (or all of them, if there are less than $k_1 + 2k_2 + 1$ such vertices). Thus at most $(k_1 + k_2)(k_1 + 2k_2 + 1)$ vertices are marked.

We show that the marked vertices satisfy the requirements. Let $H$ be a hole in $G \setminus (X, Y)$ and let $P_1$, $P_2$ be the two fragments of $H$, where $P_1$ consists of a single vertex $u \in K$. Since $H$ satisfies (*), there is a node $w_i$ corresponding to $H$. Because of the way the nodes are selected, some descendant of $w_i$ (possibly $w_i$ itself) is selected, i.e., some $w_{i_j}$ is the descendant of $w_i$. Vertex $u$ is not adjacent to any vertex of $P_2$, hence $u$ does not cover $w_i$, i.e., the tree of $u$ has nonzero distance from $w_i$. This means that the tree of $u$ has nonzero distance also from $w_{i_j}$. Consider the $k_1 + 2k_2 + 1$ vertices marked when the node $w_{i_j}$ was considered. If $u$ is not marked, then this means that there are $k_1 + 2k_2 + 1$ vertices in $K$ whose trees have not smaller distance from $w_{i_j}$, implying that these vertices do not cover $w_i$ either. At least one of these $k_1 + 2k_2 + 1$ vertices are not in $\omega(X, Y)$, let $u' \in K$ be such a vertex. Now $u'$ is not adjacent to any vertex of $P_2$, hence we can obtain a hole avoiding $u$ in $G \setminus (X, Y)$ by replacing $P_1$ with the single-vertex path consisting of $u'$ only. □

Now we are ready to prove the main lemma:

**Lemma 19** *For every $k_1, k_2$, there is a constant $c_{k_1,k_2}$ such that either we can find a necessary set or we can find an irrelevant vertex in every maximal clique of size greater than $c_{k_1,k_2}$.*

*Proof* Given a maximal clique $K$, we mark the vertices according to Lemmas 15, 16, and 18. Moreover, for each $\ell_1, \ell_2 \in F$, consider those vertices that have label $\ell_1$, but do not have label $\ell_2$, and mark $k_1 + 2k_2 + 1$ of these vertices (if there are less than $k_1 + 2k_2 + 1$ such vertices for a given $\ell_1, \ell_2$, then all of them are marked). We argue that any unmarked vertex is irrelevant. Since the number of marked vertices depends only on $k_1, k_2$, the lemma follows.

Let $u \in K$ be an unmarked vertex. To show that $u$ is irrelevant, assume that $X$ is a set of $k_1$ vertices, $Y$ is a set of $k_2$ edges, and $H$ is a hole in $G \setminus (X, Y)$ containing $u$. We have to show that $G \setminus (X, Y)$ contains a hole avoiding $u$. We construct the hole avoiding $u$ by replacing the fragment of $H$ going through $u$ with some other path going through $K$.

Let $F$, $P_1$, …, $P_s$ be the fragments of $H$. Since the paths of the fragments are independent (i.e., the vertices on two different paths are not neighbors), without loss of generality it can be assumed that $u$ is in $P_1$ and only $P_1$ intersects the clique $K$. Let $x$ and $y$ be the two end vertices of $P_1$. Path $P_1$ can contain at most one other vertex of $K$ besides $u$. We consider several cases depending on which combination of $x = y$, $u = x$, $u = y$, $|K \cap P_1| = 1$ holds (Fig. 7):

*Case 1:* $P_1$ consists of only a single vertex ($x = y = u$). Lemma 18 ensures that there is a hole in $G \setminus (X, Y)$ that does not use $u$.

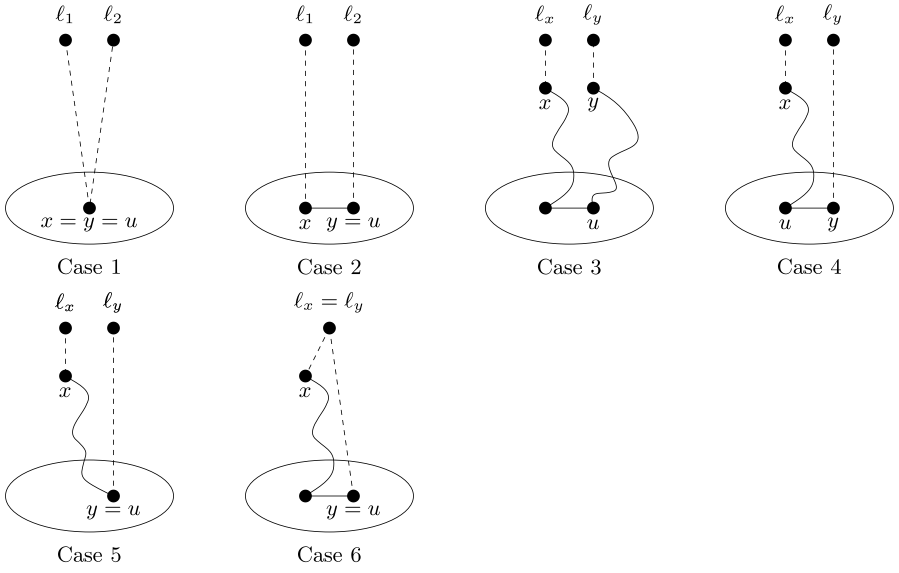

**Fig. 7** The cases in the proof of Lemma 19

In the remaining cases we assume that $x \neq y$. Moreover, without loss of generality it can be assumed that $u \neq x$. Let $\ell_x$ be the (unique) label of $x$ in $F$ and let $\ell_y$ be the (unique) label of $y$ in $F$.

*Case 2:* $P_1$ consists of two vertices $x$, $y = u$, and $P_1$ is completely contained in $K$. In this case $\ell_x \neq \ell_y$, otherwise there would be a triangle in the hole. Since $u$ is not marked, there are $k_1 + 2k_2 + 1$ marked vertices in $K$ that have label $\ell_y$ but do not have label $\ell_x$. At least one of these vertices are not in $\omega(X, Y)$, let $u'$ be such a vertex. If we replace $P_1 = \{x, u\}$ with the path $P_1' = \{x, u'\}$, then by Lemma 17 there is a hole not containing $u$.

In the remaining cases we assume without loss of generality that end point $x$ is not in $K$.

*Case 3:* $x, y \notin K$. In this case, $|K \cap P_1|$ can be either 1 or 2 (Fig. 7 sketches $|K \cap P_1| = 2$). It is possible that $\ell_x = \ell_y$ and the following proof works for that situation as well. Vertex $x$ (resp., $y$) is an $\ell_x$-dangerous (resp., $\ell_y$-dangerous) vertex with respect to $G_0 \setminus (X, Y)$ for $K$, and $u$ is a witness for that. By the way the vertices are marked (see Lemma 15) there is a marked witness $u_x$ (resp., $u_y$) in $K \setminus \omega(X, Y)$ for $x$ (resp., $y$); let $P_x$ (resp., $P_y$) be the corresponding witness path in $G_0 \setminus (X \cup u, Y)$. We consider three cases:

- $P_x \setminus x$ contains a vertex $y'$ that has label $\ell_y$. (Notice that $P_x \setminus x$ contains no vertex with label $\ell_x$, hence this case is not possible if $\ell_x = \ell_y$). Let $y'$ be the first vertex on $P_x$ (starting from $x$) with label $\ell_y$. Let $P_1'$ be the subpath of $P_x$ from $x$ to $y'$. Now $F$, $P_1'$, $P_2, \ldots, P_s$ satisfy the requirements of Lemma 17, hence $G \setminus (X, Y)$ has a hole disjoint from $u$.
- The case when $P_y \setminus y$ contains a vertex that has label $\ell_x$ follows by symmetry.
- Assume that $P_x \setminus x$ contains no vertex with label $\ell_y$ and $P_y \setminus y$ contains no vertex with label $\ell_x$. Let $P_1'$ be the path $xP_xu_xu_yP_yy$; from $u_x, u_y \in K \setminus \omega(X, Y)$ it follows that edge $u_x, u_y \notin Y$, hence $P_1'$ is fully contained in $G \setminus (X \cup u, Y)$. It is easy to see that $F$, $P_1'$, $P_2, \ldots, P_s$ satisfy the requirements of Lemma 17, hence $G \setminus (X, Y)$ has a hole disjoint from $u$.

In the remaining cases, we assume that $x \notin K$ and $y \in K$.

*Case 4:* $x \notin K$, $y \in K$, $u \neq y$ (hence $|K \cap P_1| = 2$). Vertex $x$ is an $\ell_x$-dangerous vertex for $K$, and $u$ is a witness for $x$ in $G_0 \setminus (X, Y)$. By the way the vertices are marked (see Lemma 15) there is another witness $u' \in K \setminus \omega(X, Y)$; let $P_x$ be the witness path corresponding to $u'$. Let $P_1'$ be the path $xP_xu'y$, since $u' \in K \setminus \omega(X, Y)$, the edge $u'y$ is in $G_0 \setminus (X, Y)$. Now $F$, $P_1'$, $P_2, \ldots, P_s$ satisfy Lemma 17, thus there is a hole not containing $u$.

*Case 5:* $x \notin K$, $y = u$, $\ell_x \neq \ell_y$. In this case, $|K \cap P_1|$ can be either 1 or 2 (Fig. 7 sketches $|K \cap P_1| = 1$). Vertex $x$ is an $\ell_x$-dangerous vertex for $K$, and $u$ is a witness for $x$ in $G_0 \setminus (X, Y)$. By the way the vertices are marked (see Lemma 15) there is another witness $u' \in K \setminus (X, Y)$; let $P_x$ be the witness path corresponding to $u'$. Since $u$ is not marked, there are $k_1 + 2k_2 + 1$ marked vertices in $K$ that have label $\ell_y$ but do not have label $\ell_x$. At least one of these vertices are not in $\omega(X, Y)$, let $y'$ be such a vertex. Let $P_1'$ be the path $xP_xu'y'$. Now the conditions in Lemma 17 are satisfied, hence there is a hole not containing $u$.

*Case 6:* $x \notin K$, $y = u$, $\ell_x = \ell_y$. In this case, $|K \cap P_1|$ can be either 1 or 2 (Fig. 7 sketches $|K \cap P_1| = 2$). Vertex $x$ is an $\ell_x^*$-dangerous vertex for $K$, and $u$ is a witness for $x$ in $G_0 \setminus (X, Y)$. By the way the vertices are marked (see Lemma 15) there is another witness $u' \in K \setminus \omega(X, Y)$; let $P_x$ be the witness path corresponding to $u'$. It is clear that $F$, $P_1'$ satisfy Lemma 17. □

## 6 Conclusions

We have shown that CHORDAL DELETION is fixed-parameter tractable. The problem was formulated in a way that includes both the vertex and edge deletion versions: $k_1$ vertices and $k_2$ edges have to be deleted to make the graph chordal. This formulation could be convenient for the study of other deletion problems as well. Our algorithm does not provide a problem kernel in an obvious way, thus it is a natural open question whether there is problem kernel of polynomial size for the problem.

The parameterized complexity literature contains a growing number of fixed-parameter tractability results for various deletion problems. Some of these results follow immediately from the graph minors theory of Robertson and Seymour (see [1]), while some of the results are more concrete algorithms [6, 22, 24]. Recently, a hardness result has been obtained, which shows that we cannot expect that the deletion problem is FPT for every natural graph class: Lokshtanov has shown that deleting $k$ edges/vertices to make the graph wheel-free is W[2]-hard [20]. Thus, despite the similarity of wheel-free and chordal (i.e., hole-free) graphs, the deletion problem is W[2]-hard for the former and FPT for the latter.

A natural next step would be to study the deletion problem for interval graphs. The (edge) completion problem for interval graphs was shown to be FPT by Heggernes et al. [14]. The algorithm is much more involved than chordal completion. First, all the minimal chordal completions are enumerated (using the algorithm discussed in the introduction), thus the problem is reduced to chordal graphs that are not interval graphs. The algorithm is based on a thorough understanding of such graphs. It is not clear whether a similar strategy could be used for the interval deletion problem: the algorithm presented in this paper cannot be modified such that it enumerates all the minimal solutions, in fact, it is possible that there are $n^{O(k)}$ minimal solutions. Thus it is not sufficient to solve the interval deletion problem on chordal graphs.

## References

1. Adler, I., Grohe, M., Kreutzer, S.: Computing excluded minors. In: SODA '08: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 641–650. Society for Industrial and Applied Mathematics, Philadelphia (2008)
2. Bodlaender, H.L.: A tourist guide through treewidth. Acta Cybern. **11**(1–2), 1–21 (1993)
3. Cai, L.: Fixed-parameter tractability of graph modification problems for hereditary properties. Inf. Process. Lett. **58**(4), 171–176 (1996)
4. Cai, L.: Parameterized complexity of vertex colouring. Discrete Appl. Math. **127**, 415–429 (2003)
5. Courcelle, B.: Graph rewriting: an algebraic and logic approach. In: Handbook of Theoretical Computer Science, vol. B, pp. 193–242. Elsevier, Amsterdam (1990)
6. Dehne, F., Fellows, M., Langston, M., Rosamond, F., Stevens, K.: An $O(2^{O(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem. Theory Comput. Syst. **41**(3), 479–492 (2007)
7. Dom, M., Guo, J., Hüffner, F., Niedermeier, R., Truß, A.: Fixed-parameter tractability results for feedback set problems in tournaments. In: Algorithms and Complexity. Lecture Notes in Computer Science, vol. 3998, pp. 320–331. Springer, Berlin (2006)

8. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science. Springer, New York (1999)
9. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin (2006)
10. Gallai, T.: Maximum-minimum Sätze und verallgemeinerte Faktoren von Graphen. Acta Math. Acad. Sci. Hung. **12**, 131–173 (1961)
11. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Academic Press, New York (1980)
12. Grohe, M.: Computing crossing numbers in quadratic time. J. Comput. Syst. Sci. **68**(2), 285–302 (2004)
13. Guo, J., Gramm, J., Hüffner, F., Niedermeier, R., Wernicke, S.: Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. J. Comput. Syst. Sci. **72**(8), 1386–1396 (2006)
14. Heggernes, P., Paul, C., Telle, J.A., Villanger, Y.: Interval completion with few edges. In: STOC '07: Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, pp. 374–381. ACM, New York (2007)
15. Ho, M.L.: Linear time algorithms for graphs close to chordal graphs. M. Phil. Thesis, Department of Computer Science and Engineering, The Chinese University of Hong Kong (2003)
16. Kaplan, H., Shamir, R., Tarjan, R.E.: Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. SIAM J. Comput. **28**(5), 1906–1922 (1999)
17. Kleinberg, J.: Detecting a network failure. Internet Math. **1**(1), 37–55 (2003)
18. Kloks, T.: Treewidth. Lecture Notes in Computer Science, vol. 842. Springer, Berlin (1994)
19. Lewis, J.M., Yannakakis, M.: The node-deletion problem for hereditary properties is NP-complete. J. Comput. Syst. Sci. **20**(2), 219–230 (1980)
20. Lokshtanov, D.: Wheel-free deletion is W[2]-hard. In: Proceedings of the International Workshop on Parameterized and Exact Computation (IWPEC 2008). Lecture Notes in Computer Science, vol. 5018, pp. 141–147. Springer, Berlin (2008)
21. Marx, D.: Parameterized coloring problems on chordal graphs. Theor. Comput. Sci. **351**(3), 407–424 (2006)
22. Marx, D., Schlotter, I.: Obtaining a planar graph by vertex deletion. In: 33rd International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2007). Lecture Notes in Computer Science, vol. 4769, pp. 292–303. Springer, Berlin (2007)
23. Natanzon, A., Shamir, R., Sharan, R.: Complexity classification of some edge modification problems. Discrete Appl. Math. **113**(1), 109–128 (2001)
24. Reed, B., Smith, K., Vetta, A.: Finding odd cycle transversals. Oper. Res. Lett. **32**(4), 299–301 (2004)
25. Robertson, N., Seymour, P.D.: Graph minors, XIII: the disjoint paths problem. J. Comb. Theory Ser. B **63**(1), 65–110 (1995)
26. Rose, D.J., Tarjan, R.E., Lueker, G.S.: Algorithmic aspects of vertex elimination on graphs. SIAM J. Comput. **5**(2), 266–283 (1976)
27. Yannakakis, M.: Computing the minimum fill-in is NP-complete. SIAM J. Algebr. Discrete Methods **2**(1), 77–79 (1981)