

Chordal deletion is fixed-parameter tractable

Dániel Marx

Institut für Informatik,
Humboldt-Universität zu Berlin,
Unter den Linden 6, 10099
Berlin, Germany.
`dmarx@informatik.hu-berlin.de`

Abstract. It is known to be NP-hard to decide whether a graph can be made chordal by the deletion of k vertices. Here we present a uniformly polynomial-time algorithm for the problem: the running time is $f(k) \cdot n^\alpha$ for some constant α 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 to be deleted is fixed-parameter tractable. This answers an open question of Cai [2].

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 [9]. However, it is NP-complete to decide whether a graph can be made chordal by the deletion of k vertices [6], by the deletion of k edges [7], or by the addition of k edges [10] (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* if there is an algorithm with running time $f(k) \cdot n^\alpha$, where $f(k)$ is an arbitrary function and α is a positive constant independent of k . It turns out that several NP-hard problems, such as MINIMUM VERTEX COVER and LONGEST 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. For more background, the reader is referred to the monograph of Downey and Fellows [3].

If k is a fixed constant, then the three chordal deletion/completion problems can be solved in polynomial time by complete enumeration. For example, in the vertex deletion problem we can try all the $O(n^k)$ possible sets of size k and check whether their removal makes the graph chordal. Moreover, in [1, 5] it is shown that it can be decided in $O(4^k/(k+1)^{3/2} \cdot (n+m))$ or $O(k^2nm + k^6 2^{4k})$ time whether a graph with n vertices and m edges can be made chordal by adding

k edges. 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 is also fixed-parameter tractable. This answers an open question of Cai [2].

Theorem 1. *Chordal vertex deletion is fixed-parameter tractable with parameter k , the number of vertices to be deleted.*

The *iterative compression* method introduced in [8] allows us to concentrate on the following easier problem: 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 paper is organized as follows. Section 2 reviews some basic facts on chordal graphs. In Section 3 we show how the iterative compression method of [8] can be applied to our problem. Section 4 discusses how we can reduce the size of the cliques to make our graph a bounded treewidth graph.

2 Chordal graphs

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*. Chordal graphs can be also characterized as the intersection graphs of subtrees of a tree (see e.g., [4]):

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).

For clarity, we will use the word “vertex” when we refer to the graph $G(V, E)$, and “node” when referring to $T(U, F)$. The tree T together with the subtrees T_v is called the *tree decomposition* of G . A tree decomposition of G can be found in polynomial time (see [4, 9]). We say that a vertex v contains node x if T_v contains node x . Similarly, node x can be reached from vertex v is shorthand for saying that there is a vertex v' that contains x and there is a path between v and v' . For an arbitrary node x of T , the vertices whose tree contain x induce a clique. Moreover, for every clique K , there is a node of T such that every $v \in K$ contains this node (cf. [4]). 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 in T only y and z are neighbors of x , then $T \cup x$ contains a hole of length at least 4. \square*

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

CHORDAL VERTEX DELETION

Input: A graph $G(V, E)$ and an integer k

Parameter: k

Task: Is there a set $X \subseteq V$ of size at most k such that $G \setminus X$ is chordal?

If the deletion of $X \subseteq V$ makes the graph chordal, then we say that X is a *hole cover*. It turns out that the deletion problem is very different from the edge completion problem. The algorithms in [1, 5] for chordal edge completion use the standard method of bounded search trees (with some non-trivial optimizations). The techniques rely on the fact 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.

3 Iterative compression

Reed, Smith and Vetta [8] 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* that 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 , an integer k , and a size $k + 1$ hole cover W .

Parameter: k

Task: Is there a size k hole cover X of G ?

This problem is easier: the extra input W gives us useful information on G . In particular, we know that $G \setminus W$ is chordal, our algorithm builds on this fact.

Assume that we have an algorithm with running time $f(k)n^\alpha$ for HOLE COVER COMPRESSION, then CHORDAL VERTEX DELETION can be solved as follows. 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 hole cover for each G_i . Graph G_k trivially has a size k hole cover. Now assume that G_i has a size k hole cover W . Clearly, $W \cup v_{i+1}$ is a size $k + 1$ hole cover of G_{i+1} . Therefore, the compression algorithm can be used to find a size k 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 supergraph G of G_{i+1} cannot have a size k hole cover either. The

algorithm calls the compression method at most n times, thus the total running time is $f(k)n^{\alpha+1}$, which shows that the problem is fixed-parameter tractable.

The compression algorithm will be described later in this section. Our algorithm is somewhat weaker than the one defined in the above scheme. First, it looks for only a W -avoiding hole cover, i.e., a hole cover that is disjoint from W . Furthermore, the compression algorithm either finds a W -avoiding hole cover W' , or returns a set N , whose size can be bounded by a function of k , such that every W -avoiding size k hole cover contains at least one vertex of N . (Such a set N will be called a *necessary* set.) If the compression algorithm finds that $N = \emptyset$ is necessary, then this means that there is no W -avoiding hole cover.

If the compression algorithm returns a necessary set N , then we can conclude that every size k hole cover contains at least one vertex of $N \cup W$. Therefore, we branch into $|N \cup W|$ directions: for each vertex v of $N \cup W$, we check whether there is a size $k - 1$ hole cover of $G \setminus v$. Thus the problem can be reduced to at most b_k subproblems with smaller parameter values, where b_k depends only on k . The overall algorithm is the following:

CHORDAL VERTEX DELETION(G, k)

1. Set $i := k$ and let W be the vertices of G_k .
2. *Invariant condition:* W is a size k hole cover of G_i . If $i = n$, then return “ W is a size k hole cover of G .”
3. Set $W := W \cup v_{i+1}$, now W is a size $k + 1$ hole cover of G_{i+1} .
4. Call HOLE COVER COMPRESSION(G_{i+1}, k, W).
 - If the answer is a size k hole cover W' of G_{i+1} , then let $W := W'$, $i := i + 1$, and go to Step 2.
 - If the answer is a set N , then let $T := N \cup W$.
5. For each vertex $v \in T$, call CHORDAL VERTEX DELETION($G \setminus v, k - 1$).
 - If the answer is yes for some $v \in T$, and W is a size $k - 1$ hole cover of $G \setminus v$, then answer “ $W \cup v$ is a size k hole cover of G .”
 - If the answer is no for every $v \in T$, then answer “No.”

CHORDAL VERTEX DELETION calls the compression algorithm at most n times, and may make at most b_k recursive calls to CHORDAL VERTEX DELETION with parameter $k - 1$ (where b_k is the maximum size of T). Therefore, if the compression algorithm runs in $f(k)n^\alpha$ time (which will be shown in the next section), and CHORDAL VERTEX DELETION runs in $g(k - 1)n^{\alpha+1}$ time for parameter $k - 1$, then for parameter k the algorithm runs in $g(k)n^{\alpha+1}$ time, for some appropriate constant $g(k)$. Thus by induction, we have a $g(k)n^{\alpha+1}$ time algorithm for every k , proving Theorem 1.

Now let us turn our attention to the HOLE COVER COMPRESSION algorithm itself. Assume that a size $k + 1$ hole cover W of G is given. 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 treewidth of G is at most $c - 1 + k + 1$. With standard techniques (applying Courcelle’s Theorem or dynamic programming) we can show that CHORDAL VERTEX DELETION is linear-time solvable for graphs with bounded treewidth (details omitted).

Lemma 4. *For every k and w , CHORDAL VERTEX DELETION can be solved in linear time for graphs with treewidth at most w .* \square

In Section 4, we present a method of reducing the clique size of G_0 to a constant depending only on k . A vertex $v \in V$ is *irrelevant* if every W -avoiding size k 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 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 , then the problem can be reduced to a simpler form. More precisely, for a large clique K the clique reduction algorithm does one of the following:

- *Identifies an irrelevant vertex $v \in K$.* In this case v can be deleted. If the maximum clique size is still larger than c_k , then the algorithm can be applied again. Otherwise we can use the algorithm of Lemma 4.
- *Identifies a set N of constant size such that every W -avoiding size k hole cover contains at least one vertex of N .* As mentioned above, it is possible that the compression algorithm returns a necessary set.

In the following, it is assumed that in the graph G every hole of size 4 or 5 is completely contained in W . If there is a hole H of size at most 5 that has vertices outside W , then every W -avoiding hole cover has to contain at least one vertex of $H \setminus W$, thus the compression algorithm can return $H \setminus W$ as a necessary set of constant size. Testing whether such a hole H exists can be easily done in polynomial time (e.g., by complete enumeration).

In summary, the compression algorithm makes the following steps:

HOLE COVER COMPRESSION(G, k, W)

1. By complete enumeration, determine if there is a hole H of length at most 5 that is not completely contained in W .
 - If there is such a hole H , then return “ $H \setminus W$ is a necessary set.”
2. If the clique size of $G \setminus W$ is at most c_k , then use the algorithm of Lemma 4.
3. If $G \setminus W$ has a clique K of size more than c_k , then call the clique reduction algorithm for K .
 - If the result is an irrelevant vertex v , then delete v from G , and go to Step 2.
 - If the result is a necessary set N , then return “ N is a necessary set.”

The clique reduction method is described in the following section.

4 Clique reduction

Henceforth it is assumed that $W = \{w_1, w_2, \dots, w_{k+1}\}$ is a hole cover of G . As in the previous section, let $V_0 = V \setminus W$ and denote by G_0 the chordal graph $G \setminus W$. In this section we show that if there is a large clique K in G_0 , then K contains a irrelevant vertex.

In the rest of the paper, we prove several lemmas that state certain properties of the instance. However, these properties do not always hold, but in this case the compression algorithm can identify and return a necessary set. We use the expression “We can make sure” to mean that there is a polynomial-time algorithm that finds a necessary set if the statement does not hold.

4.1 Labeling

If a vertex $v \in V_0$ is the neighbor of some vertex $\ell \in W$, then we say that v has label ℓ . A vertex can have more than one label, the labels of a given vertex form a subset of W . The following property will be used repeatedly:

Proposition 5. *If P is a path connecting u and v , vertices u and v are not neighbors, they have label ℓ , and the internal vertices of P do not have label ℓ , then every W -avoiding hole cover has to contain at least one vertex of P .*

Proof. If X is a W -avoiding hole cover disjoint from P , then $\ell u P v \ell$ contains a hole in $G \setminus X$ (Prop. 3), a contradiction. \square

In Lemma 7 we give a bound on the number of independent labeled vertices. To prove Lemma 7, we need the following technical result:

Lemma 6. *Let B be a connected set of vertices in G_0 without label t , and let A be a set of vertices with label t in the neighborhood of B . We can make sure that for every vertex $z \in B$, if z and its neighbors are deleted from G_0 , then at most $(k+1)^2$ components of the remaining graph can contain vertices from A .*

Proof. Consider the tree decomposition of G_0 , let T_v be the subtree of T that corresponds to v . Assume that after deleting z and its neighbors, components $C_1, \dots, C_{(k+1)^2+1}$ contain vertices of A , let $a_i \in A$ be in C_i . The vertices in C_i are not neighbors of z , hence their subtrees do not intersect T_z . Let p_i be the node closest to T_z that is contained in some vertex of C_i .

Let Z contain those neighbors of z that do not have label t . We show that if a vertex $v \in Z$ contains p_i for at least $k+2$ different values of i , then v is a necessary vertex. Assume without loss of generality that v contains p_1, \dots, p_{k+2} . If X is a W -avoiding size k hole cover, then without loss of generality it can be assumed that X does not contain p_1, p_2 or any of the vertices in C_1 and C_2 . Since p_1 can be reached from a_1 using only the vertices in C_1 , and v contains p_1 , thus there is a path P_1 that connects a_1 and v . The internal vertices of P_1 may have label t , let a'_1 be the t -labeled vertex of P_1 closest to v , and let P'_1 be the subpath of P_1 connecting a'_1 and v . We can define P'_2 and a'_2 in a similar way. Now $a'_1 \in C_1$ and $a'_2 \in C_2$ are not neighbors, they both have label t , and the internal vertices of the path $a'_1 P'_1 v P'_2 a'_2$ do not have label t . Hole cover X does not intersect this path, which is a contradiction by Prop. 5.

For each component C_i , there is at least one vertex of Z that contains node p_i . This follows from the fact that vertex a_i is in the neighborhood of B , thus there is a path between a_i and z whose internal vertices do not have label t .

For each component C_i , select a $z_i \in Z$ that contains p_i . We have seen in the previous paragraph that the same vertex of Z can be z_i for at most $k+1$ different values of i . Therefore, if we have more than $(k+1)^2$ components, then without loss of generality it can be assumed that z_1, \dots, z_{k+2} are distinct vertices. We claim that z is a necessary vertex. Assume that X is a W -avoiding size k hole cover that does not contain z . Without loss of generality, it can be assumed that X does not contain z_1, z_2 , or any of the vertices in C_1 and C_2 . As in the previous paragraph, there is a path P'_1 that connects a t -labeled vertex $a'_1 \in C_1$ and vertex z_1 such that only a'_1 has label t in this path. The path P'_2 is similarly defined in C_2 . Now X contains none of the vertices in the path $a'_1 P'_1 z_1 z_2 P'_2 a'_2$, thus X is not a hole cover by Prop. 5. \square

Lemma 7. *Let B be a connected subset of V_0 such that no vertex in B has label t . We can make sure that there can be at most k^4 independent vertices with label t in the neighborhood of B .*

Proof. Let $I = \{v_1, v_2, \dots, v_{k^4+1}\}$ be an independent set of vertices with label t in the neighborhood of B . We show that there is no W -avoiding size k hole cover in G . Assume that X is such a hole cover. It has to contain at least one vertex of B , otherwise $v', v'' \in I \setminus X$ can be connected by a path P whose internal vertices are in B , which is not possible by Prop. 5. For each vertex v_i we can select an $x_i \in X \cap B$ such that v_i and x_i are connected by a path P_i whose internal vertices are in $B \setminus X$. Since $|X| = k$, there has to be more than k^3 vertices v_i such that the corresponding vertices x_i are the same. Assume without loss of generality that $x_1 = x_2 = \dots = x_{k^3+1} = x$. We claim that if x and its neighbors are deleted, then vertices v_1, \dots, v_{k^3+1} are separated. By Lemma 6, is not possible (for $k \geq 3$).

Assume that v_1 and v_2 are connected by a path P that does not go through the neighborhood of x . Let y_1 (resp., y_2) be the neighbor of x on P_1 (resp., P_2). If y_1 and y_2 are the same or neighbors, then the t -labeled vertices v_1 and v_2 are connected by the walk $v_1 P_1 y_1 y_2 P_2 v_2$ in $G \setminus X$, which is not possible by Prop. 5. If y_1 and y_2 are not neighbors, then $y_1 P_1 v_1 P v_2 P_2 y_2$ is a walk connecting y_1 and y_2 such that its internal vertices are not neighbors of x . Therefore, by Prop. 3, there is a hole in G_0 , contradicting the fact that W is a hole cover. \square

4.2 Dangerous vertices

Let us fix a 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 only v has 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 the *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.

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 the t -labeled neighbor $v \in V_0$ 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. Thus 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.

First we bound by k^4 (resp., k^6) the number of independent t -dangerous (resp., t^* -dangerous) vertices. Since G_0 is chordal (hence perfect), it follows that these vertices can be covered by k^4 (resp., k^6) cliques.

Lemma 8. *We can make sure that there are at most k^4 independent t -dangerous vertices.*

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 7, the size of an independent set of t -dangerous vertices can be at most k^4 . \square

Lemma 9. *We can make sure that there are at most k^6 independent t^* -dangerous vertices.*

Proof. Consider the subgraph G'_0 of G_0 induced by the vertices without label t . Let C_1, \dots, 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 A be a set of independent t^* -dangerous vertices, and let $A_i \subseteq A$ contain a t^* -dangerous vertex v if and only if v has a witness path with internal vertices only in C_i .

If $|A_i| > k^4$, then we are ready by using Lemma 7 for connected subgraph C_i . For each t^* -dangerous vertex $v_j \in A$, fix a witness u_j and a witness path P_j . Select an arbitrary path P_j and throw away all the other paths that use the same component as P_j . Repeat this until every path is either selected or thrown away. In each step we select one path and throw away less than k^4 paths, thus $|A| > k^6$ implies that we can select more than k^2 paths. Thus assume without loss of generality that the paths P_1, \dots, P_{k^2+1} do not intersect each other outside the clique K . If a vertex $u \in K$ is contained in more than $k+1$ of these paths, then u is a necessary vertex. To see this, notice that for each $i = 1, \dots, k+1$ a W -avoiding size k hole cover has to contain at least one vertex of each P_i (Prop. 5). Since P_1, \dots, P_{k+1} intersect each other only in u , this is only possible if X contains u . Therefore, it can be assumed that every $u \in K$ is contained in at most k of the paths P_1, \dots, P_{k^2+1} . Now a simple counting argument shows that there are $k+1$ pairwise disjoint paths. This means that there is no W -avoiding size k hole cover, as it would have to intersect all these $k+1$ disjoint paths. \square

4.3 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) number of vertices in K whose deletion can make a vertex of Q non-dangerous. For every other vertex $u \in K$, if v is t -dangerous, then $v \in Q$ remains t -dangerous in $G \setminus u$. Even more is true: if X is a size k set and $v \in Q$ is t -dangerous in $G \setminus X$, then v remains t -dangerous in $G \setminus (X \cup u)$ as well.

Lemma 10. *Let Q be a clique of t -dangerous vertices. For every k , there is a constant d_k , such that we can mark d_k vertices in K such that if X is a set of size k and $v \in Q$ has an unmarked t -witness u in $G_0 \setminus X$, then v has a marked t -witness u' in $G_0 \setminus (X \cup u)$.*

Proof. Consider the tree decomposition of the chordal graph G_0 , let T_z be the subtree corresponding to a vertex z . Since Q and K are cliques, thus there are two nodes x and y such that every vertex of Q contains x , and every vertex of K contains y . Consider the unique path connecting x and y in the tree, and identify the nodes of the path with the integers $1, 2, \dots, n$, where $x = 1$ and $y = n$. Let u_1, u_2, \dots be an ordering of the vertices not having label t in K such that if a_i denotes the smallest node of T_{u_i} on this path, then the sequence a_i is non-decreasing.

We mark the vertices u_1, \dots, u_{k+1} (or up to u_j , if there are only j such vertices). Assume that the witness u of v is the vertex u_i in this ordering. If $i \leq k + 1$, then u_i is marked, and we are ready. Otherwise there is a marked vertex $u_{i'}$ for some $i' \leq k + 1 < i$ that is not contained in X . By the way the vertices are ordered, the witness path that goes from v to u_i has to go through the neighborhood of $u_{i'}$. Thus there is a path from v to $u_{i'}$, showing that $u_{i'}$ is a t -witness of v in $G \setminus (X \cup u)$. \square

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 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). The proof will appear in the full version of the paper.

Lemma 11. *Let Q be a clique of t^* -dangerous vertices. For every k , there is a constant d_k^* such that we can mark d_k^* vertices in K such that if X is a set of size k and $v \in Q$ has an unmarked t^* -witness in $G_0 \setminus X$, then v has a marked t^* -witness as well in $G_0 \setminus X$.* \square

In the next three lemmas, we extend Lemma 10 and Lemma 11 to apply not only for a clique Q of dangerous vertices, but for all dangerous vertices. Moreover, we extend it by requiring witnesses that satisfy certain other properties as well.

Let $F \subseteq W$ be a set of labels. An F -free vertex is a vertex that does not have any of the labels in F . Assume that a t -dangerous vertex v has a witness u with a corresponding witness path P . If the vertices in $P \setminus v$ are F -free, then

we say that u is an F -free witness. Moreover, let $\ell \in F$, and assume that the vertices in $P \setminus (u \cup v)$ do not have labels from F , vertex u has label ℓ but it does not have any other label from F . In this case u is said to be an ℓ -labeled F -free witness. Notice that an ℓ -labeled F -free witness is *not* an F -free witness, since it has label ℓ , which is not allowed for F -free witnesses.

By Lemma 8 and 9, there are no large independent sets of dangerous vertices. Since G_0 is chordal, it follows that the number of cliques required to cover the dangerous vertices is a constant depending only on k . The proofs of the following lemmas are based on this observation, and on the fact that number of different sets F that we have to consider depends also only on k . Details omitted.

Lemma 12. *For every k , there is a constant $c_k^{(1)}$ such that we can mark $c_k^{(1)}$ vertices in K such that for every size k vertex set X , set of labels F , and label $t \in F$, if in $G_0 \setminus X$ there is a t -dangerous vertex v with an unmarked F -free witness u , then v has an F -free marked witness u' in $G_0 \setminus (X \cup u)$.*

Lemma 13. *For every k , there is a constant $c_k^{(2)}$ such that we can mark $c_k^{(2)}$ vertices in K such that for every size k vertex set X , set of labels F , and labels $\ell, t \in F$, if in $G_0 \setminus X$ there is a t -dangerous vertex v with an unmarked ℓ -labeled F -free witness u , then v has such a marked witness in $G_0 \setminus (X \cup u)$ as well.*

Lemma 14. *For every k , there is a constant $c_k^{(3)}$ such that we can mark $c_k^{(3)}$ vertices in K such that for every size k vertex set X and label $t \in W$, if in $G \setminus X$ a t^* -dangerous vertex v has an unmarked witness u , then v has a marked witness in $G_0 \setminus (X \cup u)$ as well.*

4.4 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, \dots, P_s , the set $F = H \cap W$ together with this collection of paths will be called the *fragments* of the hole H . The internal vertices of every path P_i are F -free. Moreover, each end point has exactly one label from F . The only exception is that if a path consists of only a single vertex, then it contains exactly two labels from F . 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.

The following two easy lemmas show 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. Proofs are omitted.

Lemma 15. *Let F, P_1, \dots, P_s be the fragments of the hole H . Let P'_1 be a path that has the same end points as P_1 , and whose internal vertices are F -free. There is a hole in the graph induced by the vertices of F, P'_1, P_2, \dots, P_s .*

Lemma 16. *Let F, P_1, \dots, P_s be the fragments of the 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 ℓ_x and ℓ_y be their labels in F , respectively. Let P'_1 be a path with the following properties:*

- *the end points of P'_1 are x and y' where y' is a vertex that has label ℓ_y , but does not have any other label from F ,*
- *the internal vertices of P'_1 are F -free,*
- *if $\ell_x = \ell_y$, then x and y' are not neighbors.*

There is a hole in the graph induced by the vertices of F, P'_1, P_2, \dots, P_s .

To show that a vertex $u \in K$ is irrelevant, we have to show that every hole cover of $G \setminus (X \cup u)$ is a hole cover in $G \setminus X$. That is, if X is a size k set and there is a hole H in $G \setminus X$ going through u , then there is a hole H' in $G \setminus (X \cup u)$. The idea is to look at the fragments of H : if path P_1 is going through u , then we find a path P'_1 avoiding u , and use Lemma 15 or 16 to obtain the hole H' .

Lemma 17. *For every k , there is a constant c_k such that we can make sure that every clique of size greater than c_k contains an irrelevant vertex.*

Proof. Given a clique K , mark the vertices according to Lemma 12, 13, and 14. Moreover, for each $F \subseteq W$ and $\ell \in F$, we consider those vertices that have label ℓ , but do not have any other label from F , and we mark $k + 1$ of these vertices (if there are less than $k + 1$ such vertices for a given F and ℓ , then all of them are marked). We argue that any unmarked vertex is irrelevant. Since the number of marked vertices depends only on k , the lemma follows.

Let u be an unmarked vertex. To show that u is irrelevant, assume that $X \subseteq V_0$ has size at most k , and H is a hole in $G \setminus X$ containing u . We have to show that $G \setminus X$ contains a hole avoiding u .

Let F, P_1, \dots, 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 , let their labels be ℓ_x and ℓ_y , respectively. Path P_1 can contain at most one other vertex of K besides u . We consider several cases depending on which combination of $x = y, u = x, u = y, \ell_x = \ell_y, |K \cap P_1| = 1$ holds:

Case 1: P_1 consists of only a single vertex ($x = y = u$). Details omitted.

In the remaining cases we assume that $x \neq y$ and w.l.o.g $u \neq x$.

Case 2: P_1 consists of two vertices $x, y = u$, and P_1 is completely contained in K . Since u is not marked, there are $k + 1$ marked vertices in K that have label ℓ_y but do not have any other label from F . At least one of these vertices are not in X , let u' be such a vertex. If we replace $P_1 = \{x, u\}$ with the path $P'_1 = \{x, u'\}$, then by Lemma 16 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 . The following 4 cases handle the situation when y is in K .

Case 3: $|K \cap P_1| = 1, u = y, \ell_x \neq \ell_y$. Vertex x is an ℓ_x -dangerous vertex for K , and u is an ℓ_y -labeled F -free witness for x in $G_0 \setminus X$. By the way the

vertices are marked (see Lemma 13) there is another ℓ_y -labeled F -free witness u' in $G_0 \setminus (X \cup u)$. Let P'_1 be the witness path corresponding to u' . Now F, P'_1, P_2, \dots, P_s satisfy Lemma 16, thus there is a hole not containing u .

Case 4: $|K \cap P_1| = 2, u = y, x \notin K, \ell_x \neq \ell_y$. Since u is not marked, thus there are $k+1$ marked vertices in K that have label ℓ_y but do not have any other label from F . At least one of these vertices are not in X , let u' be such a vertex. Let P'_1 the same as P_1 , but replace the last vertex u with u' . Now Lemma 16 is satisfied, hence there is a hole not containing u .

Case 5: $|K \cap P_1| = 2, y \neq u$. Vertex x is ℓ_x -dangerous in $G_0 \setminus X$ and u is an F -free witness for x . By Lemma 12, there is a marked F -free witness u' in $G_0 \setminus (X \cup u)$; let P' be the corresponding witness path. The internal vertices of $P'_1 := xP'u'y$ are F -free, thus by Lemma 15 there is a hole in $G_0 \setminus (X \cup u)$.

Case 6: $u = y, \ell_x = \ell_y$. In this case $s = 1$: the hole consists of only $\ell_x \in W$ and the path P_1 . Vertex x is an ℓ_x^* -dangerous vertex in $G_0 \setminus X$ for K , and u is a witness for x . By the way the vertices are marked (see Lemma 14) in $G_0 \setminus (X \cup u)$ there is another witness $u' \in K$ of x . Let P'_1 be the witness path corresponding to u' . It is clear that F, P'_1 satisfies Lemma 16.

Case 7: $x, y \notin K$. Vertex x (resp., y) is an ℓ_x -dangerous (resp., ℓ_y -dangerous) vertex in $G_0 \setminus X$ for K , and u is an F -free witness for both x and y . By the way the vertices are marked (see Lemma 12) in $G \setminus X$ there is another F -free witness u'_x (resp., u'_y) for x (resp., y). Let P_x (resp., P_y) be the witness path of u'_x (resp., u'_y). Let P'_1 be the path $xP_xu'_xu'_yP_yy$ going from x to y . The internal vertices of P'_1 are F -free, thus by Lemma 15, $G \setminus X$ has a hole avoiding u . \square

References

1. L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inform. Process. Lett.*, 58(4):171–176, 1996.
2. L. Cai. Parameterized complexity of vertex colouring. *Discrete Appl. Math.*, 127:415–429, 2003.
3. R. G. Downey and M. R. Fellows. *Parameterized complexity*. Springer, 1999.
4. M. C. Golumbic. *Algorithmic graph theory and perfect graphs*. Academic Press, New York, 1980.
5. H. Kaplan, R. Shamir, and R. E. Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM J. Comput.*, 28(5):1906–1922, 1999.
6. J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *J. Comput. System Sci.*, 20(2):219–230, 1980.
7. A. Natanzon, R. Shamir, and R. Sharan. Complexity classification of some edge modification problems. *Discrete Appl. Math.*, 113(1):109–128, 2001.
8. B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.
9. D. J. Rose, R. E. Tarjan, and G. S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5(2):266–283, 1976.
10. M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Algebraic Discrete Methods*, 2(1):77–79, 1981.