Chordal Editing is Fixed-Parameter Tractable*

Yixin Cao† Dániel Marx†

June 2, 2014

Abstract

Graph modification problems are typically asked as follows: is there a small set of operations that transforms a given graph to have a certain property. The most commonly considered operations include vertex deletion, edge deletion, and edge addition; for the same property, one can define significantly different versions by allowing different operations. We study a very general graph modification problem which allows all three types of operations: given a graph G and integers k_1, k_2, and k_3, the CHORDAL EDITING problem asks whether G can be transformed into a chordal graph by at most k_1 vertex deletions, k_2 edge deletions, and k_3 edge additions. Clearly, this problem generalizes both CHORDAL VERTEX/EDGE DELETION and CHORDAL COMPLETION (also known as MINIMUM FILL-IN). Our main result is an algorithm for CHORDAL EDITING in time $2^{O(k \log k)} \cdot n^{O(1)}$, where $k := k_1 + k_2 + k_3$ and n is the number of vertices of G. Therefore, the problem is fixed-parameter tractable parameterized by the total number of allowed operations. Our algorithm is both more efficient and conceptually simpler than the previously known algorithm for the special case CHORDAL DELETION.

1 Introduction

A graph is chordal if it contains no hole, that is, an induced cycle of at least four vertices. After more than half century of intensive investigation, the properties and the recognition of chordal graphs are well understood. Their natural structure earns them wide applications, some of which might not seem to be related to graphs at first sight. During the study of Gaussian elimination on sparse positive definite matrices, Rose [23, 24] formulated the CHORDAL COMPLETION problem, which asks for the existence of a set of at most k edges whose insertion makes a graph chordal, and showed that it is equivalent to MINIMUM FILL-IN. Balas and Yu [1] proposed a heuristics algorithm for the maximum clique problem by first finding a maximum spanning chordal subgraph (see also [27]). This is equivalent to the CHORDAL EDGE DELETION problem, which asks for the existence of a set of at most k edges whose deletion makes a graph chordal. Dearing et al. [8] observed that a maximum spanning chordal subgraph can also be used to find maximum independent set and sparse matrix completion. This observation turns out to be archetypal: many NP-hard problems (coloring, maximum clique, etc.) are known to be solvable in polynomial time when restricted to chordal graphs, and hence admit a similar heuristics algorithm.

Cai [4] extended this to the exact setting. He studied the coloring problems on graphs close to certain graph classes. In particular, he asked the following question: given a chordal graph G on n vertices with k additional edges (or vertices), can we find a minimum coloring for G in $f(k) \cdot n^{O(1)}$ time? The edge version was resolved by Marx [18] affirmatively. His algorithm needs as part of the input the additional edges; to find them is equivalent to solving the CHORDAL EDGE DELETION problem. One may observe that though with slightly different purpose, the inspiration behind [1, 8] and [4] are exactly the same.

All aforementioned three modification problems , unfortunately but understandably, are NP-hard [28, 21, 14, 16]. Therefore, early work of Kaplan et al. [13] and Cai [3] focused on their parameterized complexity, and proved that that the CHORDAL COMPLETION problem is fixed-parameter tractable. Recall that a problem, parameterized by k, is *fixed-parameter tractable (FPT)* parameterized by k if there is an algorithm with runtime $f(k) \cdot n^{O(1)}$, where f is a computable function depending only on k [10]. Marx [19] showed that the complementary deletion problems, both edge and vertex versions, are also FPT. Here we consider the generalized CHORDAL EDITING problem that combines all three operations: can a graph be made chordal by deleting at most k_1 vertices, deleting at most k_2 edges, and adding at most k_3 edges. On the formulation we

*Supported by the European Research Council (ERC) grant 280152 and the Hungarian Scientific Research Fund (OTKA) grant NK105645. A preliminary version of this paper appeared in the proceedings of STACS 2014.

†Institute for Computer Science and Control, Hungarian Academy of Sciences, Email:yixin@sztaki.hu, dmarx@cs.bme.hu.

have two quick remarks. First, it does not make sense to add new vertices, as chordal graphs are hereditary (i.e., any induced subgraph of a chordal graph is chordal). Second, the budgets for different operations are not transferable, as otherwise it degenerates to CHORDAL VERTEX DELETION. Our main result establishes the fixed-parameter tractability of CHORDAL EDITING parameterized by $k := k_1 + k_2 + k_3$.

Theorem 1.1 (Main result). There is a $2^{O(k \log k)} \cdot n^{O(1)}$-time algorithm for deciding, given an n-vertex graph G, whether there are a set V_- of at most k_1 vertices, a set E_- of at most k_2 edges, and a set E_+ of at most k_3 non-edges, such that the deletion of V_- and E_- and the addition of E_+ make G a chordal graph.

As a corollary, our algorithm implies the fixed-parameter tractability of CHORDAL EDGE EDITING, which allows both edge operations but not vertex deletions—we can try every combination of k_2 and k_3 where $k_2 + k_3$ does not exceed the given bound—resolving an open problem asked by Mancini [17]. Moreover, we get a new FPT algorithm for the special case CHORDAL DELETION, and it is far simpler and faster than the algorithm of [19].

Motivation. In the last two decades, graph modification problems have received intensive attention, and promoted themselves as an independent line of research in both parameterized computation and graph theory. For graphs representing experimental data, the edge additions and deletions are commonly used to model false negatives and false positives respectively, while vertex deletions can be viewed as the detection of outliers. In this setting, it is unnatural to consider any single type of errors, while the CHORDAL EDITING problem formulated above is able to encompass both positive and negative errors, as well as outliers. We hope that it will trigger further studies on editing problems to related graph classes, especially interval graphs and unit interval graphs.

Further, since it is generally acknowledged that the study of chordal graphs motivated the theory of perfect graphs [12, 2], the importance of chordal graphs merits such a study from the aspect of structural graph theory.

Related work. Observing that a large hole cannot be fixed by the insertion of a small number of edges, it is easy to devise a bounded search tree algorithm for the CHORDAL COMPLETION problem [13, 3]. No such simple argument works for the deletion versions: the removal of a single vertex/edge suffices to break a hole of an arbitrary length. The way Marx [19] showed that this problem is FPT is to (1) prove that if the graph contains a large clique, then we can identify an irrelevant vertex whose deletion does not change the problem; and (2) observe that if the graph has no large cliques, then it has bounded treewidth, so the problem can be solved by standard techniques, such as the application of Courcelle's Theorem. In contrast, our algorithm uses simple reductions and structural properties, which reveal a better understanding of the deletion problems, and easily extend to the more general CHORDAL EDITING problem.

Of all the vertex deletion problems, we would like to single out FEEDBACK VERTEX SET, INTERVAL VERTEX DELETION, and UNIT INTERVAL VERTEX DELETION for a special comparison. Their commonality with CHORDAL VERTEX DELETION lies in the fact that the graph classes defining these problems are proper subsets of chordal graphs, or equivalently, their forbidden subgraphs contain all holes as a proper subset. All these problems admit single-exponential FPT algorithms of runtime $c^k \cdot n^{O(1)}$, where the constant c is 3.83 for FEEDBACK VERTEX SET [6], 10 for INTERVAL VERTEX DELETION [7], and 6 for UNIT INTERVAL VERTEX DELETION [25], respectively. For these problems, we can dispose of other forbidden subgraphs (i.e., triangles, small witnesses for asteroidal triples, and claws) first and their nonexistence simplifies the graph structure and significantly decrease the possible configurations on which we conduct branching (all known algorithms use bounded search trees). Interestingly, *long holes*, the main difficulty of the current paper, do not bother us at all in the three algorithms mentioned above. This partially explains why a $c^k \cdot n^{O(1)}$-time algorithm for CHORDAL VERTEX DELETION is so elusive.

Our techniques. As a standard opening step, we use the iterative compression method introduced by Reed et al. [22] and concentrate on the compression problem. Given a solution (V_-, E_-, E_+), we can easily find a set M of at most $|V_-| + |E_-| + |E_+|$ vertices such that $G - M$ is chordal. A clique tree decomposition of $G - M$ will be extensively employed in the compression step,[1] where short holes can be broken by simple branching, and the main technical idea appears in the way we break long holes. We show that a shortest hole H can be decomposed into a bounded number of segments, where the internal vertices of each segment, as well as the part of the graph "close" to them behave in a well-structured and simple way with respect to their

[1] Refer to Section 6 for more intuition behind this observation.

interaction with M. To break H, we have to break some of the segments, and the properties of the segments allow us to show that we need to consider only a bounded number of canonical separators breaking these segements. Therefore, we can branch on choosing one of these canonical separators and break the hole using it, resulting in an FPT algorithm.

Notation. All graphs discussed in this paper shall always be undirected and simple. A graph G is given by its vertex set $V(G)$ and edge set $E(G)$. We use the customary notation $u \sim v$ to mean $uv \in E(G)$, and by $v \sim X$ we mean that v is adjacent to at least one vertex in X. Two vertex sets X and Y are *completely connected* if $x \sim y$ for each pair of $x \in X$ and $y \in Y$. A hole H has the same number of vertices and edges, denoted by $|H|$. We use $N_U(v)$ as a shorthand for $N(v) \cap U$, regardless of whether $v \in U$ or not; moreover, $N_H(v) := N_{V(H)}(v)$ for a hole H. A vertex is *simplicial* if $N[v]$ induce a clique.

A set S of vertices is an x-y *separator* if x and y belong to different components in the subgraph $G - S$; it is *minimal* if no proper subset of S is an x-y *separator*. Moreover, S is a *minimal separator* if there exists some pair of x, y such that S is a minimal x-y separator. A graph is chordal if and only if every minimal separator in it induces a clique [9].

Let $\mathcal{T}$ be a tree whose vertices, called *bags*, correspond to the maximal cliques of a graph G. With the customary abuse of notation, the same symbol K is used for a bag in $\mathcal{T}$ and its corresponding maximal clique of G. Let $\mathcal{T}(x)$ denote the subgraph of $\mathcal{T}$ induced by all bags containing x. The tree $\mathcal{T}$ is a *clique tree* of G if for any vertex $x \in V(G)$, the subgraph $\mathcal{T}(x)$ is connected. It is known that the intersection of any pair of adjacent bags K and K' of $\mathcal{T}$ makes a minimal separator; in particular, it is a separator for any pair of vertices $x \in K \setminus K'$ and $y \in K' \setminus K$. A vertex is simplicial if and only if it belongs to exactly one maximal clique; thus, any non-simplicial vertex appears in some minimal separator(s) [15].

In a clique tree $\mathcal{T}$, there is a unique path between each pair of bags, and its length is called the *distance* of this pair of bags; the *distance between two subtrees* is defined to be the shortest distance between each pair of bags from these two subtrees. By definition, a pair of vertices u, v of G is adjacent if and only if $\mathcal{T}(u)$ and $\mathcal{T}(v)$ intersect. Given a pair of nonadjacent vertices u and v, there exists a unique path $\mathcal{P} = (K_u, \ldots, K_v)$ connecting $\mathcal{T}(u)$ and $\mathcal{T}(v)$, where K_u and K_v are the only bags that contain u and v respectively.

2 Outline of the algorithm

A subset of vertices is called a *hole cover* of G if its deletion makes G chordal. We say that (V_-, E_-, E_+), where $V_- \subseteq V(G)$ and $E_- \subseteq E(G)$ and $E_+ \subseteq V(G)^2 \setminus E(G)$, is a *chordal editing set* of G if the deletion of V_- and E_- and the addition of E_+, applied successively, make G chordal. Its *size* is defined to be the 3-tuple $(|V_-|, |E_-|, |E_+|)$, and we say that it is *smaller* than (k_1, k_2, k_3) if all of $|V_-| \leqslant k_1$ and $|E_-| \leqslant k_2$ and $|E_+| \leqslant k_3$ hold true and at least one inequality is strict. Note that since chordal graphs are hereditary, it does not make sense to add new vertices. The main problem studied in the paper is formally defined as follows.

CHORDAL EDITING (G, k_1, k_2, k_3)	
Input:	A graph G and three nonnegative integers k_1, k_2, and k_3.
Task:	Either construct a chordal editing set (V_-, E_-, E_+) of G that has size at most (k_1, k_2, k_3), or report that no such a set exists.

One might be tempted to define the editing problem by imposing a combined quota on the total number of operations, i.e., a single parameter $k = k_1 + k_2 + k_3$, instead of three separate parameters. However, this formulation is computationally equivalent to CHORDAL VERTEX DELETION in a trivial sense, as vertex deletions are clearly preferable to both edge operations.

We use the technique *iterative compression*: we define and solve a compression version of the problem first and argue that this implies the fixed-parameter tractability of the original problem. In the compression problem a hole cover M of bounded size is given in the input, making the problem somewhat easier: as $G - M$ is chordal, we have useful structural information about the graph. Note that the definition below has a slightly technical (but standard) additional condition, i.e., we are not allowed to delete a vertex in M.

CHORDAL EDITING COMPRESSION (G, k_1, k_2, k_3, M)	
Input:	A graph G, three nonnegative integers k_1, k_2, and k_3, and a hole cover M of G whose size is at most $k_1 + k_2 + k_3 + 1$.
Task:	Either construct a chordal editing set (V_-, E_-, E_+) of G such that its size is at most (k_1, k_2, k_3) and V_- is disjoint from M, or report that no such a set exists.

0. **return** if G is chordal or one of k_1, k_2, and k_3 becomes negative;
1. find a shortest hole H;
2. **if** H is shorter than $k + 4$ **then** guess a way to fix it; **goto** 0.
3. **else** decompose H into $O(k^3)$ segments;
   guess a segment and break it;
4. **goto** 0.

Figure 1: Outline of our algorithm for CHORDAL EDITING COMPRESSION.

Algorithm chordal-editing(G, k_1, k_2, k_3)
Input: a graph G and three nonnegative integers k_1, k_2, and k_3.
Output: a chordal editing set (V_-, E_-, E_+) of G of size at most (k_1, k_2, k_3), or "NO."

0 $i := 0$; $V_- := \emptyset$; $E_- := \emptyset$; $E_+ := \emptyset$;
1 **if** $i = n$ **then return** (V_-, E_-, E_+).
2 $X := V_- \cup \{v_{i+1}\}$ and one endpoint (picked arbitrarily) from each edge in $E_- \cup E_+$;
3 **for each** X_- of X of size $\leqslant k_1$ **do**
3.1 **call** Theorem 2.1 with $(G^{i+1} - X_-, k_1 - |X_-|, k_2, k_3, X \setminus X_-)$; *$M := X \setminus X_-$ is the modulator.*
3.2 **if** the answer is (V'_-, E'_-, E'_+) **then**
$(V_-, E_-, E_+) := (V'_- \cup X_-, E'_-, E'_+)$;
$i := i + 1$; **goto** 1;
4 **return "NO."** *no subset X_- works in step 3.*

Figure 2: Algorithm for CHORDAL EDITING.

The hole cover M is called the *modulator* of this instance. We use $k := k_1 + k_2 + k_3$ to denote the total numbers of operations. The main part of this paper will be focused on an algorithm for CHORDAL EDITING COMPRESSION. Its outline is described in Figure 1. We will endeavor to prove the following theorem.

Theorem 2.1. CHORDAL EDITING COMPRESSION is solvable in time $2^{O(k \log k)} \cdot n^{O(1)}$.

Let us briefly explain here steps 1 and 2 of the algorithm for CHORDAL EDITING COMPRESSION, while leaving the main technical part, step 3, for later sections. We can find in time $O(n^3(n + m))$ a shortest hole H as follows: we guess three consecutive vertices $\{v_1, v_2, v_3\}$ of H, and then search for the shortest v_1-v_3 path in $G - (N[v_2] \setminus \{v_1, v_3\})$. In order to destroy a hole H, we need to perform at least one of the possible $|V(H) \setminus M|$ vertex deletions (vertices in M are avoided here), $|H|$ edge deletions, or $O(|H|^2)$ edge insertions that affect H. Therefore, if the length of H is no more than $k + 3$, then we can fix it easily by branching into $O(k^2)$ direction. Hence we may assume $|H| \geqslant k + 4 > k_3 + 3$. Such a hole cannot be fixed with edge additions only; thus at least one deletion has to occur on this hole. As we shall see in Section 3, the hole can be divided into a bounded number of "segments" (paths), of which at least one needs to be "broken." In our case, breaking a segment means more than deleting one vertex or edge from it, and it needs a strange mixed form of separation: we have to separate two vertices by removing both edges and vertices. We study this notion of mixed separation on chordal graphs in Section 4. Finally, we show in Section 5 that there is a bounded number of canonical ways of breaking a segment and we may branch on choosing one segment and one of the canonical ways of breaking it. This completes the proof of Theorem 2.1, which enables us to prove Theorem 1.1.

Proof of Theorem 1.1. Let $v_1, \ldots, v_n$ be an arbitrary ordering of $V(G)$, and let G^i be the subgraph induced by the first i vertices. Note that $G^n = G$. The algorithm described in Figure 2 iteratively finds a chordal editing set of G^i from $i = 1$ to n; the solution for G^i is used in solving G^{i+1}. The algorithm maintains as an invariant that (V_-, E_-, E_+) is a chordal editing set of size at most (k_1, k_2, k_3) of G^i for the current i. For each G^i, note that $|X| \leqslant k + 1$, step 3 generates at most $2^{O(k)}$ instances of CHORDAL EDITING COMPRESSION, each with parameter at most (k_1, k_2, k_3), and thus can be solved in $2^{O(k \log k)} \cdot n^{O(1)}$ time. There are n iterations, and the total runtime of the algorithm is thus $2^{O(k \log k)} \cdot n^{O(1)}$. □

3 Segments

We need to define a hierarchy of vertex sets V_0, V_1, and V_2. Each set is a subset of the preceding one, and all of them induce chordal subgraphs. Let A denote the set of common neighbors of the shortest hole H found in step 1 (Figure 1), and define $A_M = A \cap M$ and $A_0 = A \setminus M$. We can assume that A induces a clique: if two vertices $x, y \in A$ are nonadjacent, then together with two nonadjacent vertices v_1 and v_3 of H, they form a 4-hole xv_1yv_3x. The following observation follows from the fact that H is the shortest hole of G.

Proposition 3.1. A vertex not in A is adjacent to at most three vertices of H and these vertices have to be consecutive in H.

The first set is defined by $V_0 = V(G) \setminus (M \cup A)$, and let $G_0 = G[V_0]$. Note that $\{M, A_0, V_0\}$ partitions $V(G)$, and H is disjoint from A_0. Since $|H| \geqslant k+4 > |M|$ and G_0 is chordal, the hole H intersects both M and V_0. Every component of $H - M$ is an induced path of G_0, and there are at most $|M|$ such paths. We divide each of these paths into $O(k^2)$ parts; observing $|M| = O(k)$, this leads to to a decomposition of H into $O(k^3)$ segments. Let P denote such a path $v_1v_2 \dots v_p$ in H, where $v_i \in V_0$ for $1 \leqslant i \leqslant p$ and the other neighbors of v_1 and v_p in H (different from v_2 and v_{p-1} respectively) are in M. We restrict our attention to paths with $p > 3$ (as there is a trivial bound for shorter paths). For such paths, Proposition 3.1 implies that the distance between v_1 and v_p in G_0 is at least 3. A further consequence is $v_1 \not\sim v_p$.

Let us fix a clique tree $\mathcal{T}$ for the chordal subgraph G_0. We take the unique path $\mathcal{P}$ of bags $K_1, \dots, K_q$ that connects the disjoint subtrees $\mathcal{T}(v_1)$ and $\mathcal{T}(v_p)$ in $\mathcal{T}$, where $K_1 \in \mathcal{T}(v_1)$ and $K_q \in \mathcal{T}(v_p)$. The condition $p > 3$ implies that $q > 2$. The removal of K_1 and K_q will separate $\mathcal{T}$ into a set of subtrees, one of which contains all K_ℓ with $1 < \ell < q$; let $\mathcal{T}_1$ denote this nonempty subtree. The second set, V_1, is defined to be the union of all bags in $\mathcal{T}_1$ and $\{v_1, v_p\}$. By definition and observing that V_1 fully contains P, it induces a connected subgraph.

We then focus on bags in $\mathcal{P}$ and their union. (One may have judiciously observed that these vertices induce an interval graph.) From the definition of clique tree, we can infer that v_1 and v_p appear only in K_1 and K_q respectively, while every internal vertex of P appears in more than one bags of $\mathcal{P}$. For every i with $1 \leqslant i \leqslant p$, we denote by $\texttt{first}(i)$ (resp., $\texttt{last}(i)$) the smallest (resp., largest) index ℓ such that $1 \leqslant \ell \leqslant q$ and $v_i \in K_\ell$, e.g., $\texttt{first}(1) = \texttt{last}(1) = \texttt{first}(2) = 1$ and $\texttt{last}(p-1) = \texttt{first}(p) = \texttt{last}(p) = q$. As P is an induced path, for each i with $1 < i < p$, we have

$$\texttt{first}(i) \leqslant \texttt{last}(i-1) < \texttt{first}(i+1) \leqslant \texttt{last}(i). \tag{1}$$

For $1 \leqslant \ell < q$, we define $S_\ell = K_\ell \cap K_{\ell+1}$. For any pair of nonadjacent vertices v_i, v_j in P, (i.e., $1 \leqslant i < i+1 < j \leqslant p$,) all minimal v_i-v_j separators are then $\{S_\ell \mid \texttt{last}(i) \leqslant \ell < \texttt{first}(j)\}$.

The third set, V_2, is defined to be the union of vertices in all induced v_1-v_p paths in G_0. Note that V_2 and A_0 are completely connected: given a pair of nonadjacent vertices $x \in V_2$ and $y \in A_0$, we can find a hole of $G - M$ that consists of y and part of a v_1-v_p path through x in G_0. Since a vertex x is an internal vertex of an induced v_1-v_p path of G_0 if and only if it is in some minimal v_1-v_p separator of G_0, we have (noting $q > 2$)

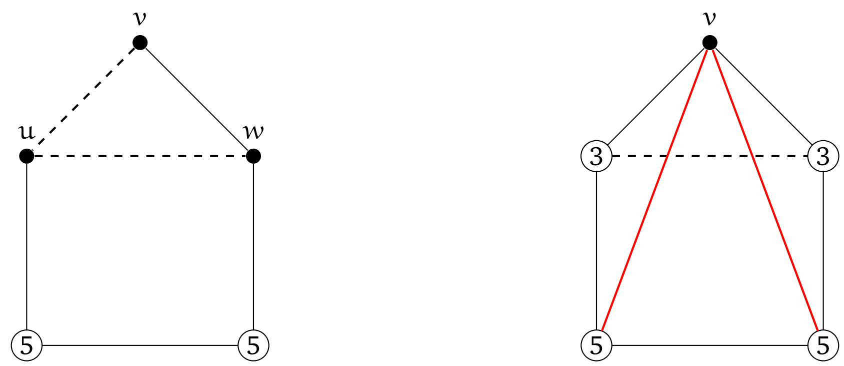
Proposition 3.2. A vertex is in $V_2 \setminus \{v_1, v_p\}$ if and only if it appears in more than one bags of $\mathcal{P}$. Moreover, $V_2 \setminus \{v_1, v_p\} \subseteq \bigcup_{1<\ell<q} K_\ell$.

The definitions of V_0 and G_0 depend upon the hole H, while the definitions of V_1 and V_2 depend upon both the hole H and the path P. In this paper, the hole H will be fixed, and we are always concerned with a particular path of H, which will be specified before the usage of V_1 and V_2.

The set $V_0 \setminus V_1$ is easily understood, and we now consider $V_1 \setminus V_2$. Given a pair of nonadjacent vertices $x, y \in V_2$, we say that x lies to the *left* (resp., *right*) of y if the bags of $\mathcal{P}$ containing x have smaller (resp., greater) indices than those containing y. If an induced path of $G[V_2]$ consists of three or more vertices, then its endvertices are nonadjacent and have a left-right relation. This relation can be extended to all pairs of consecutive (and adjacent) vertices x, y in this path, the one with smaller distance to the left endvertex of the path is said *to the left of the other*. It is easy to verify that these two definitions are compatible.

Lemma 3.3. For any component C of the subgraph induced by $V_1 \setminus V_2$, the set $N_{V_0}(C)$ induces a clique and there exists ℓ such that $1 < \ell < q$ and $N_{V_0}(C) \subseteq K_\ell$.

Proof. Consider a vertex $x \in C$, which is different from v_1 and v_p. Since $x \in V_1$, it appears in some bag of $\mathcal{T}_1$. Recall that the only bag of $\mathcal{T}_1$ that is adjacent to K_1 is K_2. We argue first that $x \notin K_1$: recall that V_1 is disjoint from $K_1 \setminus (\{x\} \cup K_2)$, and thus if $x \in K_1$ then it has to be in K_2 as well, but then $x \in V_2$ (Proposition 3.2), contradicting that $C \subseteq V_1 \setminus V_2$. For the same reason, $x \notin K_q$. As a result, $N_{V_0}(x) \subseteq V_1$, and then $N_{V_0}(C) \subseteq V_2$. It now suffices to show that $N_{V_0}(C)$ induces a clique. Suppose for contradiction that there is a pair of



(a) deleting edges (dashed) incident to v (b) adding edges (red) incident to v

Figure 3: Possible modifications to a simplicial vertex v. ($\textcircled{x}$ means a clique of x vertices and an edge means all the edges between the two cliques/vertices.) (a) A minimal solution with two edge deletions. (b) A minimal solution with one edge deletion and two edge addition.)

nonadjacent vertices $x, y \in N_{V_0}(C)$. We can find an induced v_1-v_p path P' through x and y; without loss of generality, let x lie to the left of y, i.e., $P' = v_1 \cdots x \cdots y \cdots v_p$. Let x' and y' be the first and last vertices in P' that are adjacent to C, and let $x'P''y'$ be an induced path with all internal vertices from C. Note that x' either is x or lies to the left of x in P' and y' either is y or lies to the right of y, which imply $x' \not\sim y'$. Thus $v_1 \cdots x'P''y' \cdots v_p$ is an induced v_1-v_p path through C, which is impossible. This completes the proof. □

Such a component C is called a *branch* of P, and we say that it is *near to* $v_i \in P$ if there is an ℓ with $\texttt{first}(i) \leqslant \ell \leqslant \texttt{last}(i)$ satisfying the condition of Lemma 3.3. In other words, C is near to $v_i \in P$ if and only if $N_{V_0}(C) \subseteq N[v_i]$. Applying Proposition 3.1 on any vertex in $N_{V_0}(C)$, we conclude that a branch is near to at most three vertices of P. If there exists some hole passing through C, then C has to be adjacent to M: by Lemma 3.3 and recalling that V_2 and A_0 are completely connected, $N_{V_0}(C) \cup A_0$ is a clique, and thus a hole cannot enter and leave C both via $N_{V_0}(C) \cup A_0$. The converse is not necessarily true: some branch that is adjacent to M might still be disjoint from all holes, e.g., if $N(C)$ is a clique. This observation inspires us to generalize the definition of simplicial vertices to sets of vertices.

Definition 1. A set X of vertices is called *simplicial in a graph* G if $N[X]$ induces a chordal subgraph of G and $N(X)$ induces a clique of G.

It is easy to verify that a simplicial set of vertices is disjoint from all holes. This suggests that simplicial sets are irrelevant to CHORDAL EDITING problem and we may never want to add/delete edges incident to a vertex in a simplicial set. However, this is not true: as Figure 3 shows, we may need to add/delete such edges if $N(X)$ was modified. As characterized by the following lemma, this is the only reason for touching it in the solution. In other words, a simplicial set X will only concern us after $N(X)$ has been changed. We say that a chordal editing set (V_-, E_-, E_+) *edits* a set $X \subset V(G)$ of vertices if either V_- contains a vertex of X or $E_- \cup E_+$ contains an edge with at least one endpoint in X. We use a classic result of Dirac [9] stating that the graph obtained by identifying two cliques of the same size from two chordal graphs is also chordal.

Lemma 3.4. A minimal chordal editing set edits a simplicial set U only if it removes at least one edge induced by $N(U)$.

Proof. Let (V_-, E_-, E_+) be a minimal editing set of G such that E_- does not contain any edge induced by $N(U)$. We restrict the editing set to the subgraph $G - U$, i.e., we consider the set $(V_- \setminus U, E_- \setminus (U \times V(G)), E_+ \setminus (U \times V(G)))$, and let G' be the graph obtained by applying it to G. Clearly $G' - U = G - U$ is chordal, where $N(U) \setminus V_-$ induces a clique. Also chordal is the subgraph of G' induced by $N[U] \setminus V_-$. Both of them contain the clique $N(U) \setminus V_-$. Since G' can be obtained from them by identifying $N(U) \setminus V_-$, it is also chordal. Then by the minimality of (V_-, E_-, E_+), it must be the same as $(V_- \setminus U, E_- \setminus (U \times V(G)), E_+ \setminus (U \times V(G)))$, and this proves this lemma. □

Now we are ready to define segments of the path P, which are delimited by some special vertices called junctions. By definition, a branch is simplicial in G_0, but not necessarily simplicial in G.

Definition 2 (Segment)**.** A vertex $v \in P$ is called a *junction (of* P*)* if

(1) some bag K that contains v is adjacent to $M \setminus A_M$;

(2) some branch near to v is adjacent to $M \setminus A_M$;

(3) some branch near to v is not simplicial in G; or

(4) $N_{V_2}(v)$ is not completely connected to A.

A sub-path $v_s \cdots v_t$ of P is called a *segment*, denoted by $[v_s, v_t]$, if v_s and v_t are the only junctions in it.

We point out that the four types are not exclusive, and one junction might be in more than one types. For a junction v of type (1) or (2), we say that the vertex in $M \setminus A_M$ used in its definition *witnesses* it. Let us briefly explain the intuition behind the definition of junctions and segments.

Remark 3.5. For a junction v of type (1) or (2), there is a connection from v to $M \setminus A_M$ that is *local* to v in some sense; for a junction v of type (3) or (4), there is a hole near to v, and its disposal might interfere with that of H. On the other hand, since there is no junctions inside a segment $[v_s, v_t]$, if another hole H' intersects it, then H' has to "go through the whole segment." Or precisely, H' necessarily enters and exits the segment via $N[v_s]$ and $N[v_t]$, respectively.

The definition of junction and segment extends to all paths of $H - M$. In polynomial time, we can construct V_0 for H and V_1, V_2 for each path P of $H - M$, from which all junctions of H can be identified. In particular, the endvertices of P are adjacent to $M \setminus A_M$, hence junctions (of type (1)). As a result, every vertex in $V(H) \setminus M$ is contained in some segment, and in each path of $H - M$, the number of segments is the number of junctions minus one.

We are now ready for the main result of this section that gives a cubic bound on the number of segments of H. It should be noted the constants—both the exponent and the coefficient—in the following statement are not tight, and the current values simplify the argument significantly. Recall that a vertex not in A sees at most three vertices in H, and they have to be consecutive.

Theorem 3.6. If H contains more than $|M| \cdot (12k^2 + 87k + 75)$ segments, then we can either find a vertex that has to be in V_-, or return "NO."

Proof. We show that H contains at most $|M| \cdot (12k^2 + 87k + 75)$ junctions. Recall that there are at most $|M|$ paths in $H - M$. To obtain a contradiction, we suppose that some path P of $H - M$ contains $12k^2 + 87k + 75$ junctions. Let us first attend to junctions of type (1) in P.

Claim 1. Each $w \in M \setminus A_M$ witness at most 14 junctions of type (1) in P.

Proof. We are proving a stronger statement of this claim, i.e., w witness at most 14 junctions of type (1) in the entire hole H. Suppose, for contradiction, that 15 vertices in H appear in some bag adjacent to w; let X be this set of vertices. Assume first that X is consecutive. At most 3 of them are adjacent to w, and they are consecutive in H. Thus, we can always pick 6 consecutive vertices from X that are disjoint from $N_H(w)$; let them be $\{v_i, \ldots, v_{i+5}\}$. By definition, there are two vertices $u_1, u_2 \in V_0 \cap N(w)$ such that $u_1 \sim v_i$ and $u_2 \sim v_{i+5}$. It is easy to verify that $u_2 \not\sim v_{i+2}$ and $u_1 \not\sim v_{i+3}$ and $u_1 \not\sim u_2$. Therefore, we can find an induced u_1-u_2 path with all interval vertices from $\{v_i, \ldots, v_{i+5}\}$. The length of this path is at least 3, and hence it makes a hole with w of length at most 9. Assume now that X is not consecutive in P, then we can pick a pair of nonadjacent vertices v_i, v_j from X such that the $v_\ell \notin X$ for every $i < \ell < j$. There are two vertices $u_1, u_2 \in V_0 \cap N(w)$ such that $u_1 \sim v_i$ and $u_2 \sim v_j$. It is easy to verify that $w u_1 v_i \cdots v_j u_2 w$ is a hole. By assumption that $|X| \geqslant 15$, we have $j - i \leqslant |H| - 13$. In either case, we end with a hole strictly shorter than H. The contradictions prove this claim. ⌟

Claim 2. If some vertex $w \in M \setminus A_M$ witnesses $5k + 75$ junctions of types (1) and (2) in P, then we can return "NO."

Proof. Let X be this set of junctions, we order them according to their indices in P and group each consecutive five from the beginning. We omit groups that contain junctions of type (1) witnessed by w, and in each remaining group, we pair the second and last vertices in it. According to Claim 1, we end with at least $k + 1$ pairs, which we denote by $(v_{\ell_1}, v_{r_1}), \cdots, (v_{\ell_{k+1}}, v_{r_{k+1}}), \cdots$.

For each pair (v_{ℓ_j}, v_{r_j}), where $1 \leqslant j \leqslant k + 1$, we construct a hole H_j as follows. By definition, there is a branch C_{ℓ_j} (resp., C_{r_j}) whose neighborhood in H is a proper subset of $\{v_{\ell_j - 1}, v_{\ell_j}, v_{\ell_j + 1}\}$ (resp., $\{v_{r_j - 1}, v_{r_j}, v_{r_j + 1}\}$). By the selection of the pair v_{ℓ_j} and v_{r_j} (two vertices of X have been skipped in between), they are nonadjacent, and $r_j - \ell_j > 2$. Therefore, C_{ℓ_j} and C_{r_j} are distinct and necessarily nonadjacent.

Since C_{ℓ_j} induces a connected subgraph and is adjacent to both w and $\{v_{\ell_j-1}, v_{\ell_j}, v_{\ell_j+1}\}$, we can find an induced w-v_{ℓ_j+1} path P_{ℓ_j} with all internal vertices from $C_{\ell_j} \cup \{v_{\ell_j-1}, v_{\ell_j}\}$. Likewise, we can obtain an induced w-v_{r_j-1} path P_{r_j} with all internal vertices from $C_{r_j-1} \cup \{v_{r_j}, v_{r_j+1}\}$. These two paths P_{ℓ_j} and P_{r_j}, together with $v_{\ell_j+1} \dots v_{r_j-1}$, make the hole H_j: we have $\ell_j + 1 < r_j - 1$; for each $\ell_j + 1 \leqslant s \leqslant r_j - 1$, $v_s \not\sim w$; and for each $\ell_j + 1 < s < r_j - 1$, $v_s \not\sim C_{\ell_j}, C_{r_j}$. This hole goes through w. This way we can construct $k+1$ holes, and it can be easily verified that they intersect only in w. Since we are not allowed to delete w, we cannot fix all these holes by at most k operations. Thus we can return "NO." ⌟

If Claim 2 applies, then we are already done; otherwise, there are at most $|M| \cdot (5k + 74)$ junctions of the first two types in P. We proceed by considering the set B of junctions that are only of type (3) or (4) but not of the first two types. Its number is at least (noting $|M| \leqslant k + 1$)

$$(12k^2 + 87k + 75) - (5k + 74) \cdot |M| \geqslant 7k^2 + 7k + 1.$$

We order B according to their indices in P, and let b_i denote the index of the ith vertex of B in P. For each $0 \leqslant i \leqslant k(k+1)$, we use the $(7i+3)$th vertex of B to construct a hole H_i. Then we argue that this collection of holes either allows us to identify a vertex that has to be in the solution, or conclude infeasibility.

The first case is when $v_{b_{7i+3}}$ is of type (4): there is a pair of nonadjacent vertices $x \in N_{V_2}(v_{b_{7i+3}})$ and $y \in A$. In this case we can assume that x is adjacent to neither $v_{b_{7i+1}}$ nor $v_{b_{7i+5}}$; otherwise $x v_{b_{7i+1}} y v_{b_{7i+3}} x$ or $x v_{b_{7i+3}} y v_{b_{7i+5}} x$ is a 4-hole, which contradicts the fact that H is the shortest. In other words, x only appears in some bag between $K_{\mathtt{last}(b_{7i+1})}$ and $K_{\mathtt{first}(b_{7i+5})}$; on the other hand, by definition of V_2, it appears in at least two of these bags. There is thus an induced $v_{b_{7i+1}}$-$v_{b_{7i+5}}$ path P_i via x in $G[V_2]$. Starting from x, we traverse P_i to the left until the first vertex x_1 that is adjacent to y; the existence of such a vertex is ensured by the fact that $y \sim v_{b_{7i+1}}$. Similarly, we find the first neighbor x_2 of y in P_i to the right of x. Then the sub-path of P_i between x_1 and x_2, together with y, gives the hole H_i. By construction, no vertex of $H_i - y$ is adjacent to $v_{b_{7i}}$ or $v_{b_{7i+6}}$.

In the other case, $v_{b_{7i+3}}$ is type (4): some branch C_i near to $v_{b_{7i+3}}$ is not simplicial in G. By definition, either the subgraph induced by $N(C_i)$ is not a clique, or the subgraph induced by $N[C_i]$ is not chordal. Since $v_{b_{7i+3}}$ does not satisfy the conditions of type (1) and (2), $N(C_i) \cap M \subseteq A_M$, i.e., $N(C_i) \setminus V_0 \subseteq A$. On the other hand, according to Lemma 3.3, $N(C_i) \cap V_0$ induces a clique. Therefore, there must be a pair of nonadjacent vertices $x \in N(C_i) \cap V_0$ and $y \in A_M$. As C_i is near to $v_{b_{7i+3}}$, it must hold that $x \in N(v_{b_{7i+3}})$; this has already been discussed in the previous case. Suppose now that $N(C_i)$ induces a clique and there is a hole H_i in $N[C_i]$. We have seen that $N[C_i] \cap M = A_M$, thus this hole H_i intersects A_M; let w be a vertex in $V(H_i) \cap A_M$. If H_i is disjoint from A_0, then no vertex in $H_i \setminus M$ can be adjacent to $v_{b_{7i}}$ or $v_{b_{7i+5}}$. Otherwise, it contains some vertex $u \in A_0$; noting that A induces a clique, $H_i \cap A = \{u, w\}$. Moreover, $N(C_i) \cap V_2$ is in the neighborhood of $v_{b_{7i+3}}$ and therefore $N(C_i) \cap V_2$ and $N(C_j) \cap V_2$ are disjoint for $i \neq j$: the existence of a vertex $x \in V_2$ adjacent to both C_i and C_j would contradict Proposition 3.1 (noting that the distance of $v_{b_{7i+3}}$ and $v_{b_{7j+3}}$ is greater than 2 on the hole H).

In sum, we have a set $\mathcal{H}$ of at least $k(k+1)+1$ distinct holes such that (1) each hole in $\mathcal{H}$ contains at most one vertex of A_0, and (2) the intersection of any pair of them is in A. Recall that each hole has length at least $k+4$, hence cannot be fixed by edge additions only. If there is a $u \in A_0$ contained in at least $k+1$ holes of $\mathcal{H}$, then we have to put u into V_-; otherwise we have to delete distinct elements (edges or vertices) to break different holes, which is impossible. Now assume that no such a vertex u exists, then there must be $k+1$ holes that intersect only in M, which allow us to return "NO." □

4 Mixed separators in chordal graphs

Given a pair of nonadjacent vertices x, y of a graph, we say that a pair of vertex set V_S and edge set E_S is a *mixed* x-y *separator* if the deletion of V_S and E_S leaves x and y in two different components; its size is defined to be $(|V_S|, |E_S|)$. A mixed x-y separator is *inclusion-wise minimal* if there exists no other mixed x-y separator (V_S', E_S') such that $V_S' \subseteq V_S$ and $E_S' \subseteq E_S$ and at least one containment is proper. If (V_S, E_S) is an inclusion-wise minimal mixed x-y separator in graph F, then each component of $F - V_S - E_S$ is an induced subgraph of F. Therefore, we have the following characterization of inclusion-wise minimal mixed separators in chordal graphs.

Proposition 4.1. In a chordal graph, all components obtained by deleting an inclusion-wise minimal x-y separator are chordal.

Consider an inclusion-wise minimal x-y separator (V_S, E_S) in a chordal graph F. Let $\mathcal{T}^F$ be a clique tree of F. The degenerated case where $E_S = \emptyset$ is well understood: V_S itself makes an x-y separator. If $E_S \neq \emptyset$, then in the path that connects $\mathcal{T}^F(x)$ and $\mathcal{T}^F(y)$, at least one bag is disconnected by the deletion of V_S and E_S. This bag contains at most $|V_S| + |E_S| + 1$ vertices. On the other hand, the remaining vertices of every bag K, i.e., $K \setminus V_S$, appear in either one or two components of $F - V_S - E_S$. In the latter case, the two components are precisely that contain x and y, respectively; otherwise the mixed separator cannot be inclusion-wise minimal.

Lemma 4.2. Let x and y be a pair of nonadjacent vertices in a chordal graph F. For any pair of nonnegative integers (a, b), we can find a mixed x-y separator of size at most (a, b) or asserts its nonexistence in time $3^{a+b+1} \cdot |V(F)|^{O(1)}$.

```
Algorithm mixed-separator(F, x, y, a, b)
INPUT: a chordal graph F, nonadjacent vertices x and y, and nonnegative integers a and b.
OUTPUT: a mixed x-y separator (V_S, E_S) of size at most (a, b) or "NO."

0     find a minimum vertex x-y separator S;
      if |S| ⩽ a then return (S, ∅).
1     X = ∅;   Y = ∅;   Z = ∅;
2     build a clique tree T^F for F;
      guess a bag K from the path of bags connecting T^F(x) and T^F(y);
3     enqueue(Q, K);
4     while Q ≠ ∅ do
4.1       K = dequeue(Q);
4.2       if |K \ (X ∪ Y ∪ Z)| > a − |Z| + b − |E(F) ∩ (X × Y)| + 1 then return "NO";
4.3       guess a partition (X_K, Y_K, Z_K) of K \ (X ∪ Y ∪ Z);
4.4       X = X ∪ X_K;   Y = Y ∪ Y_K;   Z = Z ∪ Z_K;
4.5       if a < |Z| or b < |E(F) ∩ (X × Y)| then return "NO";
4.6       for each bag K′ adjacent to K that is not "processed" do
              if K′ intersects both X and Y then enqueue(Q, K′);
4.7       mark K "processed";
5     V_S = Z;   E_S = E(F) ∩ (X × Y).
6     if x and y are disconnected in F − V_S − E_S then return (V_S, E_S);
          else return "NO."
```

Figure 4: Algorithm finding mixed separators in chordal graphs.

Proof. We use the algorithm described in Figure 4. If the size of minimum x-y separators is no more than a, then step 0 will give a correct separator, and hence main part of the algorithm looks for a solution with $E_S \neq \emptyset$. Let us explain the variables used in the algorithm and formally state its invariants. The algorithm processes bags one by one, and maintains a partition (X, Y, Z) of vertices in all bags that have been processed. The partition can be arbitrary if there exists no mixed x-y separator of the designated size. Otherwise the partition satisfies for some mixed x-y separator (V_S^*, E_S^*) of the designated size that (1) X and Y are in the same components of $F - V_S^* - E_S^*$ as x and y respectively; and (2) $Z \subseteq V_S^*$. The queue $\mathcal{Q}$ keeps all bags to be processed, and a bag is enqueued if it intersect both X and Y. A bag to be processed must be adjacent to a previously process bag, and since the queue starts from a single bag, at the end of the algorithm, all processed bags induce a connected subtree of $\mathcal{T}^F$.

The algorithm has no false positives. Therefore, to verify its correctness, we show that each inclusion-wise minimal mixed x-y separator (V_S, E_S) of size at most (a, b) can be found. We initialize $\mathcal{Q}$ by guessing a bag in the path connecting $\mathcal{T}^F(x)$ and $\mathcal{T}^F(y)$ that is disconnected by the deletion of (V_S, E_S); the existence of such a bag follows from previous discussion. Main work of the algorithm is done in the loop of step 4, each iteration of which processes a bag in $\mathcal{Q}$. Let K be the bag under processing. By assumption, if a vertex $v \in K \setminus (X \cup Y \cup Z)$ is not in V_S^*, then it has to be incident to an edge in E_S^*. Let $b' = |K \setminus (X \cup Y \cup Z)| - (a - |Z|)$; then at least b' vertices of K will remain in $F - V_S^* - E_S^*$, and any nontrivial partition of it has at least $b' - 1$ edges (when one side has precisely one vertex). It cannot exceed $b - |E(F) \cap (X \times Y)|$; this justifies the exit condition 4.2. Steps 4.3–4.5 are straightforward. Step 4.6 enqueues bags that have to be separated by the deletion of (V_S^*, E_S^*).

It remains to verify that (V_S, E_S) constructed in step 5 is the objective mixed separator, i.e., $V_S^* = V_S = Z$ and $E_S^* = E_S = E(F) \cap (X \times Y)$. Since we have shown that $Z \subseteq V_S^*$ and $E(F) \cap (X \times Y) \subseteq E_S^*$, and by assumption, x (resp., y) remains connected to X (resp., Y) in $F - V_S - E_S$, it suffices to show that X and Y are disconnected in $F - V_S - E_S$. Suppose for contradiction that there is an induced path P connecting $v_x \in X$ and $v_y \in Y$ in $F - V_S - E_S$. Let P be the path $u_1 \cdots u_p$ where $u_1 = v_x$ and $u_p = v_y$. Without loss of generality, assume that all internal vertices of P are disjoint from $X \cup Y$. Let l be the smallest index such that $1 < l < p$ and $u_l \sim Y$. We argue that $u_l \sim X$ as well. Otherwise, let l' be the largest index such that $1 < l' < l$ and $u_{l'} \sim X$. It is easy to verify that in $F - V_S$, subgraphs induced by X, Y, and $X \cup Y$ are all connected. Hence we can find an induced $u_{l'}$-u_l path with all internal vertices in $X \cup Y$; this path and $u_{l'} \cdots u_l$ make a hole, which is impossible as F is chordal. Let $v'_x \in X$ and $v'_y \in Y$ be neighbors of u_l. Note that all bags handled in step 4 induce a connected subtree of $\mathcal{T}^F$, and in particular, it intersects both $\mathcal{T}^F(v'_x)$ and $\mathcal{T}^F(v'_y)$. If $v'_x \sim v'_y$, then there is a bag containing $\{u_l, v'_x, v'_y\}$. Let us focus on bags that contain v'_x and v'_y. At least one of such bags is separated, and all of them are then enqueued in concession. If $v'_x \not\sim v'_y$, then u_l is in any v'_x-v'_y separator, and at least one bag that contains u_l is handled. In both cases, u_l has to be in $X \cup Y \cup Z$. This gives a contradiction, and hence (V_S, E_S) must be a mixed x-y separator. This completes the proof of the correctness.

We now analyze the runtime. In step 2, there are at most $|V|$ bags in the path connecting $\mathcal{T}^F(x)$ and $\mathcal{T}^F(y)$, and thus the bag K can be found in $O(|V|)$ time. Note that this step is run only once. The only step that takes exponential time is 4.3. The set $K \setminus (X \cup Y \cup Z)$ has $3^{|K \setminus (X \cup Y \cup Z)|}$ partitions, and after each execution of step 4.3, the budget decreases by at least $|K \setminus (X \cup Y \cup Z)| - 1$. In total, this is upper bounded by 3^{a+b+1}. This completes the proof. □

We remark that the problem of finding a mixed separator of certain size is fixed-parameter tractable even in general graphs: the treewidth reduction technique of Marx et al. [20] can be used after a simple reduction (subdivide each edge, color the new vertices red and the original vertices black, and find a separator with at most k_1 black vertices and at most k_2 red vertices). However, the algorithm of Lemma 4.2 for the special case of chordal graphs is simpler and much more efficient.

The definition of mixed separator can be easily generalized to two disjoint vertex sets—we may simply shrink each set into a single vertex and then look for a mixed separator for these two new vertices. Another interpretation of Lemma 4.2 is the following.

Corollary 4.3. Let X and Y be a pair of nonadjacent and disjoint sets of vertices in a chordal graph F. For any nonnegative integer $a \leqslant k_1$, in time $3^{k_1+k_2+1} \cdot |V(F)|^{O(1)}$ we can find the minimum number b such that $b \leqslant k_2$ and there is a mixed X-Y separator of size (a, b) or assert that there is no mixed X-Y separator of size (a, k_2).

5 Proof of Theorem 2.1

We are now ready to put everything together and finish the analysis of the algorithm. We say that a chordal editing set is minimum if there exists no chordal editing set with a smaller size. Note that a segment is contained in a unique path of $H - M$, which determines V_1 and V_2.

Proof of Theorem 2.1. Let (V^*_-, E^*_-, E^*_+) be a minimum chordal editing set of G of size no more than (k_1, k_2, k_3). We start from a closer look at how it breaks H; by Theorem 3.6, we may assume that H contains $O(k^3)$ segments. There are three options for breaking H. In the first case, V^*_- contains some junction, or E^*_- contains some edge of H that is in $M \times V_0$. In this case, we can branch on including one of these vertices or edges into the solution; there are $O(k^3)$ of them. Otherwise, we need to delete an internal vertex or edge from some segment. Let $d = 2k + 4$. In the second case, we delete either (1) a vertex that is at distance at most d (on the cycle) from a junction; or (2) an edge whose both endpoints are at distance at most d (on the cycle) from a junction. In particular, this case must apply when we are breaking a segment of length at most $2d$. If one of the two aforementioned cases is correct, then we can identify one vertex or edge of the solution by branching. In total, there are $O(k^4)$ branches we need to try.

Henceforth, we assume that none of these two cases holds. We still have to delete at least one vertex or edge from H; this vertex or edge must belong to some segment $[v_s, v_t]$ with $t - s > 2d$. This is the third case, where we use $s' = s + d$ and $t' = t - d$. Recall that any segment $[v_s, v_t]$ belongs to some maximal path P of $H - M$, on which V_1 and V_2 are well defined. For any pair of indices i, j with $s \leqslant i < i + 3 \leqslant j \leqslant t$, we use $U_{[i,j]}$ to denote the union of the set of bags in the nonempty subtree of $\mathcal{T} - \{K_{\texttt{last}(i)}, K_{\texttt{first}(j)}\}$ that contains $\{K_{\texttt{last}(i)+1}, \ldots, K_{\texttt{first}(j)-1}\}$, plus the two vertices v_i and v_j. Let $G_{[i,j]}$ be the subgraph induced by $U_{[i,j]}$.

Claim 3. There must be some segment $[v_s, v_t]$ with $t - s > 2d$ such that vertices $v_{s'}$ and $v_{t'}$ are disconnected in $G_{[s,t]} - V^*_- - E^*_-$.

Proof. We prove by contradiction. Consider first a segment $[v_s, v_t]$ with $t - s > 2d$. Suppose for contradiction that $v_{s'}$ and $v_{t'}$ are connected in $G_{[s,t]} - V^*_- - E^*_-$. We can find an induced $v_{s'}$-$v_{t'}$ path $P_{[s',t']}$ in $G_{[s,t]} - V^*_- - E^*_-$, which has to visit every bag K_ℓ with $\texttt{last}(s') \leqslant \ell \leqslant \texttt{first}(t')$. Appending to it $v_s \cdots v_{s'}$ and $v_s \cdots v_{s'}$, we get a v_s-v_t path $P_{[s,t]}$ in $G_{[s,t]} - V^*_- - E^*_-$. From $P_{[s,t]}$ we can extract an induced v_s-v_t path $P'_{[s,t]}$ of $G_{[s,t]} - V^*_- - E^*_-$. It is also a v_s-v_t path of $G_{[s,t]}$, where the distance between v_s and v_t is $t - s > 2d$, and thus the length of $P'_{[s,t]}$ is larger than $2d > 2k_3 + 4$. On the other hand, a segment $[v_s, v_t]$ of length at most $2d$ remains intact in $G - V^*_- - E^*_-$ by assumption, which can be used as the v_s-v_t path.

We have then obtained for each segment $[v_s, v_t]$ of H an induced v_s-v_t path $P'_{[s,t]}$ in $G - V^*_- - E^*_-$. Concatenating all these paths, as well as edges of H in $M \times V(G)$, we get a closed walk C. To verify that C is a hole, it suffices to verify that the internal vertices of $P'_{[s,t]}$ is disjoint and nonadjacent to other parts of C. On the one hand, no internal vertex of $P'_{[s,t]}$ is adjacent to $M \setminus A_M$ by definition (C is disjoint from A). On the other hand, all internal vertices of $P'_{[s,t]}$ appear in the subtree that contains $K_{\mathtt{last}(s+4)}$ in $\mathcal{T} - \{K_{\mathtt{last}(s+3)}, K_{\mathtt{first}(t-3)}\}$, while no vertex in the v_t-v_s path in C does. This verifies that C is a hole of $G - V^*_- - E^*_-$. Since the length of C is longer than $2k_3 + 4$, it cannot be made chordal by the addition of the at most k_3 edges of E^*_+. This contradiction proves the claim. ⌟

In other words, there is a segment $[v_s, v_t]$ such that (V^*_-, E^*_-) contains some inclusion-wise minimal mixed $\{v_s, \ldots, v_{s'}\}$-$\{v_{t'}, \ldots, v_t\}$ separator (V^*_S, E^*_S) in $G_{[s,t]}$. The resulting graph obtained by deleting (V^*_S, E^*_S) from $G_{[s,t]}$ is characterized by the following claim.

Claim 4. Let (V_S, E_S) be an inclusion-wise minimal mixed $v_{s'}$-$v_{t'}$ separator in $G_{[s',t']}$, and let $G' = G - V_S - E_S$. Let X be the component of $G' - (K_{\mathtt{last}(i)} \cup A)$ for some i with $s \leqslant i \leqslant s'$ that contains $v_{s'}$. Then X is simplicial in G'.

Proof. By definition, $N_{G'}(X) \subseteq K_{\mathtt{last}(i)} \cup A$ and is a clique in G; otherwise v_{i+1} must be a junction of type (3), which is impossible. Since (V_S, E_S) is inclusion-wise minimal, no edge in E_S is induced by $N_{G'}[X]$. In particular, $N_{G'}(X)$ induces the same subgraph in G and G', which is a clique. It remains to show that $N_{G'}[X]$ induces a chordal subgraph of G'. A vertex in $N_{G'}[X]$ is either in V_2, some branch, or A. For every branch C near to some vertex v_i with $s < i < t$, $C \cap N_{G'}[X]$ is simplicial. On the other hand, by definition of segments, $V_2 \cap N_{G'}[X]$ is completely connected to A. Therefore, $N_{G'}[X]$ induces a chordal subgraph in G'. ⌟

A symmetric claim holds for the other side of the segment $[v_s, v_t]$. That is, for any i with $t' \leqslant i \leqslant t$, the component X of $G' - (K_{\mathtt{last}(i)} \cup A)$ that contains $v_{t'}$ is simplicial in G'. We now consider the subgraph obtained from G by deleting (V^*_S, E^*_S), i.e., $G' = G - V^*_S - E^*_S$. Note that $(V^*_- \setminus V^*_S, E^*_- \setminus E^*_S, E^*_+)$ is a minimum chordal editing set of G'.

Claim 5. For any mixed $\{v_s, \ldots, v_{s'}\}$-$\{v_{t'}, \ldots, v_t\}$ separator (V^*_S, E^*_S) of size at most $(|V^*_S|, |E^*_S|)$ in $G_{[s,t]}$, substituting (V_S, E_S) for (V^*_S, E^*_S) in (V^*_-, E^*_-, E^*_+) gives another minimum editing set to G.

Proof. We first argue the existence of some vertex $v_{s''}$ with $s \leqslant s'' \leqslant s'$ such that E^*_- contains no edge induced by $K_{\mathtt{last}(s'')}$. For each s'' with $s \leqslant s'' \leqslant s'$, since $\mathtt{last}(s'') \geqslant \mathtt{first}(s''+1)$ and every vertex in them is adjacent to at most 3 vertices of H (Proposition 3.1), bags $K_{\mathtt{last}(s'')}$ and $K_{\mathtt{last}(s''+2)}$ are disjoint. In particular, an edge cannot be induced by both $K_{\mathtt{last}(s'')}$ and $K_{\mathtt{last}(s''+2)}$. Suppose that E^*_- contains an edge induced by $K_{\mathtt{last}(s'')}$ for each s'' with $s \leqslant s'' < s'$, then we must have $|E_-| > (s' - s)/2 \geqslant k_2$, which is impossible. Likewise, we have some vertex $v_{t''}$ with $t' \leqslant t'' \leqslant t$ such that E^*_- contains no edge induced by $K_{\mathtt{first}(t'')}$. By Claim 4, it follows that every vertex of $U_{[s'',t'']}$ is in a simplicial set of $G - V^*_S - E^*_S$. Since $(V^*_- \setminus V^*_S, E^*_- \setminus E^*_S, E^*_+)$ is a minimum chordal editing set to $G - V^*_S - E^*_S$, we have by Lemma 3.4 that $(V^*_- \setminus V^*_S, E^*_- \setminus E^*_S, E^*_+)$ does not edit any vertex of $U_{[s'',t'']}$.

Suppose that there is a hole C in the graph obtained by applying $((V^*_- \setminus V^*_S) \cup V_S, (E^*_- \setminus E^*_S) \cup E_S, E^*_+)$ to G. By construction, C contains a vertex of $U_{[s',t']} \subseteq U_{[s'',t'']}$. However, by Claim 4, every vertex of $U_{[s'',t'']}$ is in some simplicial set of $G - V_S - E_S$ and, as $(V^*_- \setminus V^*_S, E^*_- \setminus E^*_S, E^*_+)$ does not edit $U_{[s'',t'']}$, every such vertex is in a simplical set after applying $((V^*_- \setminus V^*_S) \cup V_S, (E^*_- \setminus E^*_S) \cup E_S, E^*_+)$ to G. Thus no vertex of $U_{[s'',t'']}$ is on a hole, a contradiction. ⌟

For any segment $[v_s, v_t]$, we can use Corollary 4.3 to find all possible sizes of a minimum mixed $\{v_s, \ldots, v_{s'}\}$-$\{v_{t'}, \ldots, v_t\}$ separator. There are at most k_1 of them. By Claim 5, one of them can be used to compose a minimum chordal editing set. In each iteration, we branch into $O(k^4)$ instances to break a hole, and in each branch decreases k by at least 1. The runtime is thus $O(k)^{4k} \cdot n^{O(1)} = 2^{O(k \log k)} \cdot n^{O(1)}$. This completes the proof. □

6 Concluding remarks

We have presented the first FPT algorithm for the general modification problem to a graph class that has infinite number of obstructions. It is natural to ask for its parameterized complexity on other related graph classes, especially for those classes on which every single-operation version is already known to be FPT. The most interesting candidates include unit interval graphs and interval graphs. The fixed-parameter tractability of their completion versions were shown by Kaplan et al. [13] and Villanger et al. [26]; their vertex deletion versions were shown by van 't Hof and Villanger [25] and Cao and Marx [7]. A very recent result of Cao [5] complemented them by showing that the edge deletion versions are FPT as well.

We would like to draw attention to the similarity between CHORDAL DELETION and the classic FEEDBACK VERTEX SET problem, which asks for the deletion of at most k vertices to destroy all *cycles* in a graph, i.e., to make the graph a forest. The ostensible relation is that the forbidden induced subgraphs of forests are precisely all holes and triangles. But triangles can be easily disposed of and its nonexistence significantly simplifies the graph structure. On the other hand, each component of a chordal graph can be represented as a clique tree, which gives another way to be correlate these two problems.

Recall that vertices with degree less than two are irrelevant for FEEDBACK VERTEX SET, while degree two vertices can also be preprocessed, and thus it suffices to consider graphs with minimum degree three. Earlier algorithms for FEEDBACK VERTEX SET are based on some variations of the upper bounds of Erdős and Pósa [11] on the length of shortest cycles in such a graph. For CHORDAL VERTEX DELETION, our algorithm can be also interpreted in this way. First of all, a simplicial vertex participates in no holes, and thus can be removed safely.

Reduction 1. Remove all simplicial vertices.

Note that a simplicial vertex corresponds to a leaf in the clique tree, Reduction 1 can be viewed as a generalization of the disposal of degree-1 vertices for FEEDBACK VERTEX SET. For FEEDBACK VERTEX SET, we "smoothen" a degree-2 vertex by removing it and adding a new edge to connect its two neighbors. This operation shortens all cycles through this vertex and result in an equivalent instance. To have a similar reduction rule, we need an explicit clique tree,[2] so we consider the compression problem, which, given a hole cover M, asks for another hole cover M' disjoint from M. The following reduction rule will only be used after Reduction 1 is not applicable, then no vertex inside a segment can have a branch. Let S_ℓ denote the separator $K_\ell \cap K_{\ell+1}$ in the clique tree.

Reduction 2. Let $[v_s, v_t]$ be a segment and $|S_\iota| = \min_{\mathtt{last}(s) \leqslant i < \mathtt{first}(t)} |S_i|$. If there exists S_ℓ such that S_ℓ is disjoint from $K_{\mathtt{last}(s)} \cup K_{\mathtt{first}(t)}$ and there exists $v \in S_\ell \setminus S_\iota$, then remove v and insert edges to make $N(v)$ a clique.

After both reductions are exhaustively applied, we can use an argument similar as Theorem 3.6 to show that either the length of a shortest hole is $O(k^4)$ or there is no solution. However, unlike FEEDBACK VERTEX SET, Reductions 1 and 2 do not directly imply a polynomial kernel for CHORDAL VERTEX DELETION. Therefore, we leave it open the existence of polynomial kernels for the CHORDAL VERTEX DELETION problem and its compression variation.

References

[1] Egon Balas and Chang Sung Yu. Finding a maximum clique in an arbitrary graph. *SIAM Journal on Computing*, 15(4):1054–1068, 1986.

[2] Claude Berge. Some classes of perfect graphs. In F. Harary, editor, *Graph Theory and Theoretical Physics*, pages 155–166. Academic Press, New York, 1967.

[3] Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.

[4] Leizhen Cai. Parameterized complexity of vertex colouring. *Discrete Applied Mathematics*, 127(3):415–429, 2003.

[5] Yixin Cao. Linear recognition of almost (unit) interval graphs. arXiv:1403.1515, 2014.

[2] This can be surely extended to some local clique tree structure, and we use clique tree here for simplicity.

[6] Yixin Cao, Jianer Chen, and Yang Liu. On feedback vertex set: New measure and new structures. In Haim Kaplan, editor, *Proceedings of the 12th Scandinavian Workshop on Algorithm Theory, SWAT 2010*, volume 6139 of *LNCS*, pages 93–104. Springer, 2010.

[7] Yixin Cao and Dániel Marx. Interval deletion is fixed-parameter tractable. In Chandra Chekuri, editor, *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 122–141. SIAM, 2014. Full version avalaible as arXiv:1211.5933.

[8] P.M. Dearing, D.R. Shier, and D.D. Warner. Maximal chordal subgraphs. *Discrete Applied Mathematics*, 20(3):181–190, 1988.

[9] Gabriel A. Dirac. On rigid circuit graphs. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 25(1):71–76, 1961.

[10] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Springer, 1999.

[11] Paul Erdős and Lajos Pósa. On the maximal number of disjoint circuits of a graph. *Publicationes Mathematicae Debrecen*, 9:3–12, 1962.

[12] András Hajnal and Janos Surányi. Über die auflösung von graphen in vollständige teilgraphen. *Annales Universitatis Scientarium Budapestinensis de Rolando Eötvös Nominatae Sectio Mathematica*, 1:113–121, 1958.

[13] Haim Kaplan, Ron Shamir, and Robert E. Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM Journal on Computing*, 28(5):1906–1922, 1999. A preliminary version appeared in FOCS 1994.

[14] Mukkai S. Krishnamoorthy and Narsingh Deo. Node-deletion NP-complete problems. *SIAM Journal on Computing*, 8(4):619–625, 1979.

[15] John G. Lewis, Barry W. Peyton, and Alex Pothen. A fast algorithm for reordering sparse matrices for parallel factorization. *SIAM Journal on Scientific and Statistical Computing*, 10(6):1146–1173, November 1989.

[16] John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980. Preliminary versions independently presented in STOC 1978.

[17] Federico Mancini. *Graph modification problems related to graph classes*. PhD thesis, University of Bergen, Bergen, Norway, 2008.

[18] Dániel Marx. Parameterized coloring problems on chordal graphs. *Theoretical Computer Science*, 351(3):407–424, 2006.

[19] Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010.

[20] Dániel Marx, Barry O'Sullivan, and Igor Razgon. Finding small separators in linear time via treewidth reduction. *ACM Transactions on Algorithms*, 9(4):30.1–30.35, 2013. A preliminary version appeared in STACS 2010.

[21] Assaf Natanzon, Ron Shamir, and Roded Sharan. Complexity classification of some edge modification problems. *Discrete Applied Mathematics*, 113(1):109–128, 2001. A preliminary version appeared in LNCS vol. 1665, WG 1999.

[22] Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.

[23] Donald J. Rose. Triangulated graphs and the elimination process. *Journal of Mathematical Analysis and Applications*, 32(3):597–609, 1970.

[24] Donald J. Rose. A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. In Ronald C. Reed, editor, *Graph Theory and Computing*, pages 183–217. Academic Press, New York, 1973.

[25] Pim van 't Hof and Yngve Villanger. Proper interval vertex deletion. *Algorithmica*, 65(4):845–867, 2013.

[26] Yngve Villanger, Pinar Heggernes, Christophe Paul, and Jan Arne Telle. Interval completion is fixed parameter tractable. *SIAM Journal on Computing*, 38(5):2007–2020, 2009. A preliminary version appeared in STOC 2007.

[27] Jue Xue. Edge-maximal triangulated subgraphs and heuristics for the maximum clique problem. *Networks*, 24(2):109–120, 1994.

[28] Mihalis Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 2(1):77–79, 1981.