Chordal Editing is Fixed-Parameter Tractable*

Yixin Cao[†‡] Dániel Marx[†]

Abstract

Graph modification problems typically ask for 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 that 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 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

Chordal graphs are arguably the oldest and most important perfect graph class [12, 2, 3]. A graph is chordal if very cycle of length larger than three has a chord. Chordal graphs have many nice structural properties, which earn them wide applications. Balas and Yu [1] proposed a heuristic algorithm for the maximum clique problem by first finding a maximum spanning chordal subgraph (see also [25]). 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 approximate maximum independent sets 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 heuristic algorithm. Some applications of chordal graphs 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 addition makes a graph chordal, and showed that it is equivalent to MINIMUM FILL-IN.

Cai [5] extended this observation to the exact setting. He studied the coloring problems on graphs close to certain graph classes. In particular, he asked the following question: given an n-vertex graph G that can be obtained from a chordal graph by adding k 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. Likewise, to decide whether

*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 (MTA SZTAKI), Budapest, Hungary. dmarx@cs.bme.hu.

‡Department of Computing, Hong Kong Polytechnic University, Hong Kong, China. yixin.cao@polyu.edu.hk.

a graph G can be obtained from a chordal graph by k additional vertices is equivalent to solving the CHORDAL VERTEX DELETION problem. One may observe that though with slightly different purpose, the inspirations behind [1, 8, 25] and [5] are exactly the same.

All three aforementioned modification problems, unfortunately but understandably, are NP-hard [26, 21, 14, 16]. Therefore, early work of Kaplan et al. [13] and Cai [4] focused on their parameterized complexity, and proved that CHORDAL COMPLETION is fixed-parameter tractable (FPT). Recall that a graph problem, with a nonnegative parameter k, is FPT if there is an algorithm solving it in time $f(k) \cdot n^{O(1)}$, where f is a computable function depending only on k and n is the number of vertices in the input graph [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: the task is to decide whether a graph can 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 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 chordal.

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 Marx [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 algorithmic 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. 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. Chordal graphs contain no holes (i.e., induced cycles of at least four vertices) as induced subgraphs. Observing that a large hole cannot be fixed by the addition of a small number of edges, it is easy to devise a bounded search tree algorithm for the CHORDAL COMPLETION problem [13, 4]. 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 the deletion problems are 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 those to forests (also known as FEEDBACK VERTEX SET), interval graphs, and unit interval graphs for a special comparison. Their commonality with CHORDAL VERTEX DELETION lies in the fact that these graph classes are proper subsets of chordal graphs, or equivalently, their forbidden subgraphs contain all holes as a proper subset. For these problems, we can dispose of those small forbidden subgraphs first and their nonexistence simplifies the graph structure and significantly decreases the possible configurations on which we conduct branching (all known algorithms use bounded search trees). As a result, each of them admits a $c^k \cdot n^{O(1)}$-time algorithms for some small constant c. However, *long holes*, which do not bother us at all in these three problems, turn out to be the main difficulty of the current paper. 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_+) to a graph G, 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 hole H of the minimum length 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 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 them. 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. The vertex set and edge set of a graph G are denoted by $V(G)$ $E(G)$ respectively. We use the customary notation $u \sim v$ to mean $uv \in E(G)$, and by $v \sim U$ we mean that v is adjacent to at least one vertex in U. Two vertex sets X and Y are *completely connected* if $x \sim y$ for every 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 $N[v]$ induces 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*. Note that the definition of x-y separators requires them to be disjoint from $\{x, y\}$. Minimal separators and induced paths are connected by the following fact.

Proposition 1.2. A vertex is an internal vertex of an induced u-v path if and only if it is in some minimal u-v separator.

Let $\mathcal{T}$ be a tree whose vertices, called *bags*, correspond to the maximal cliques of a connected 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. We say that $\mathcal{T}$ is a *clique tree* of G if for every $x \in V(G)$, all bags containing x induce a subtree of $\mathcal{T}$, denoted by $\mathcal{T}(x)$. It is known that a connected graph is chordal if and only if it has a clique tree, which has at most n bags [9]. Between any pair of adjacent bags K and K' of a clique tree $\mathcal{T}$, the intersection $K \cap K'$ is a minimal x-y separator for any pair of vertices $x \in K \setminus K'$ and $y \in K' \setminus K$. In our setting, this separator is necessarily nonempty. 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]. By definition, a pair of vertices u, v of G is adjacent if and only if $\mathcal{T}(u)$ and $\mathcal{T}(v)$ intersect. Otherwise, there exists a unique path $K_u \dots K_v$ connecting $\mathcal{T}(u)$ and $\mathcal{T}(v)$ in $\mathcal{T}$, where K_u and K_v are the only bags that contain u and

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

v respectively; the intersection of any pair of consecutive bags in this path is a (not necessarily minimal) u-v separator.

A notational remark is worth here. For technical reasons, we have restricted clique trees in this paper to be defined only on connected chordal graphs, or components in a general chordal graph. However, in literature it is more common that clique trees are defined without the condition on connectedness. Indeed, given a set of clique trees for all components of a disconnected chordal graph, one may add arbitrary edges to connect them into a single tree, where some adjacent bags may share no common vertices. We remark also that clique trees of chordal graphs are not unique, and our algorithm does not rely on a particular one.

2 Outline of the algorithm

A subset of vertices is called a *hole cover* of a graph G if its deletion makes G chordal. Let $V_- \subseteq V(G)$, and let E_- and E_+ be a set of edges and a set of non-edges of $G - V_-$ respectively. We say that (V_-, E_-, E_+) is an *chordal editing set* of G if the deletion of E_- from and the addition of E_+ to $G - V_-$ create a chordal graph. 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 set exists.

We use $k := k_1 + k_2 + k_3$ to denote the total numbers of operations. One might be tempted to define the editing problem by imposing a combined quota on the total number of operations 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*, i.e., 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 a bounded size is given in the input. Note that the definition below has a slightly technical (but standard) additional condition: we are not allowed to delete any vertex from 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 $M \subseteq V(G)$ of G with $\|M\| \leqslant k_1 + k_2 + k_3 + 1$.
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.

The hole cover M is called the *modulator* of this instance, which makes the problem somewhat easier: as $G - M$ is chordal, we have useful structural information about the graph. The main part of this paper will be focused on an algorithm for CHORDAL EDITING COMPRESSION. More specifically, we will endeavor to prove the following theorem.

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

Roughly speaking, our algorithm for CHORDAL EDITING COMPRESSION either repetitively calls the following steps or reduces the instance when it identifies a vertex that has to be in V_-.

1. find a shortest hole H;

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."

$V_- \leftarrow \emptyset$; $E_- \leftarrow \emptyset$; $E_+ \leftarrow \emptyset$;
for $i = 1, \ldots, n$ **do**
1. $X \leftarrow V_- \cup \{v_i\}$ and any endvertex from each edge in $E_- \cup E_+$;
2. **branch** on each $X_- \subseteq X$ of size $\leqslant k_1$ **do**
   $G^i \leftarrow G[\{v_1, \ldots, v_i\}]$;
   call Theorem 2.1 with $(G^i - X_-, k_1 - |X_-|, k_2, k_3, X \setminus X_-)$;
   if the answer is (V'_-, E'_-, E'_+) **then** $(V_-, E_-, E_+) \leftarrow (V'_- \cup X_-, E'_-, E'_+)$;
   else terminates this branch;
3. **if** all branches have been terminated **then return "NO"**;
   else $i \leftarrow i + 1$;

return any (V_-, E_-, E_+) in a successful branch of the last iteration.

Figure 1: Algorithm for CHORDAL EDITING.

2. if H is shorter than $k + 4$, then guess a way to fix it;
3. otherwise, decompose H into $O(k^3)$ segments, and guess a segment and break it.

Let us briefly explain here steps 1 and 2, while leaving the main technical part, step 3, for later sections. A shortest hole H can be detected in time $O(n^3(n + m))$ 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 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 $O(k^2)$ directions. Hence we may assume $|H| \geqslant k + 4 > k_3 + 3$ in step 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 the main theorem.

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 1 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 main work is done in the for-loop, which 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. The set X found in step 1 contains no more than $k + 1$ vertices, and then step 2 generates at most $2^{O(k)}$ instances of CHORDAL EDITING COMPRESSION, where $X \setminus X_-$ is the modulator M. Each instance has 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 running time of the algorithm is thus $2^{O(k \log k)} \cdot n^{O(1)}$. □

3 Segments

We consider the CHORDAL EDITING COMPRESSION problem. Let H be the shortest hole we have found, which is assumed to be longer than $k+3$. We denote by A the set of common neighbors of H, and define $A_M = A \cap M$ and $A_0 = A \setminus M$. We can assume that A induces a clique: if two vertices in A are nonadjacent, then they form a 4-hole together with any two nonadjacent vertices of H. The following observation follows from the fact that H is the shortest hole of G.

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

Let $V_0 = V(G) \setminus (M \cup A)$. For each component in the chordal subgraph $G_0 = G[V_0]$, we fix a clique tree. 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 a decomposition of H into $O(k^3)$ parts. For this purpose, it suffices to consider paths longer than $k+3$. Let P denote such a path $v_1 v_2 \dots v_p$ in H, then $v_i \in V_0$ 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.

It is worth noting that the definition of V_0 depends upon the hole H. We shall now define two more vertex sets V_1 and V_2, which depend upon, apart from H, the sub-path P we are considering, and the clique tree $\mathcal{T}$ for the component of G_0 containing P.

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 and K_q are the only bags of $\mathcal{P}$ that contain v_1 and v_p respectively. The condition $p > k+3$ implies that $q > 2$. The removal of K_1 and K_q from $\mathcal{T}$ 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 set V_1 is defined to be the union of all bags in $\mathcal{T}_1$ and $\{v_1, v_p\}$. Since these bags induce a subtree, and our definition of clique tree requires a nonempty intersection between two adjacent bags, V_1 is a subset of V_0 and induces a connected subgraph.

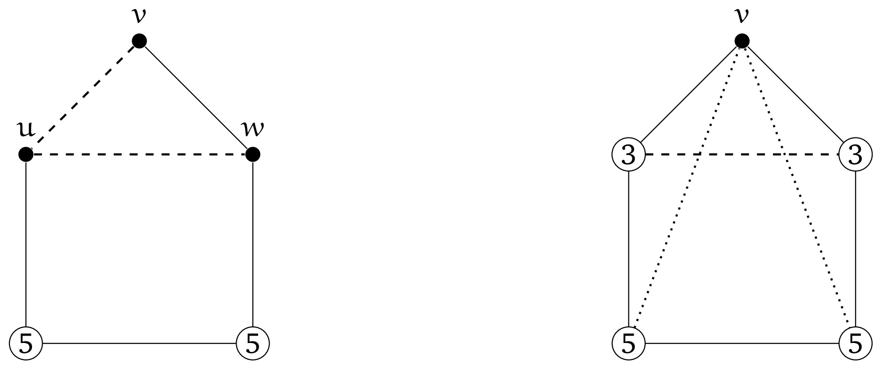
We then focus on bags in $\mathcal{P}$ and their union. (One may have judiciously observed that these vertices induce an interval subgraph.) For every i with $1 \leqslant i \leqslant p$, we denote by $\texttt{first}(i)$ (resp., $\texttt{last}(i)$) the smallest (resp., largest) index ℓ such that $1 \leqslant \ell \leqslant q$ and $v_i \in K_\ell$. Recall that v_1 and v_p appear only in K_1 and K_q respectively, hence $\texttt{first}(1) = \texttt{last}(1) = \texttt{first}(2) = 1$ and $\texttt{last}(p-1) = \texttt{first}(p) = \texttt{last}(p) = q$. On the other hand, every internal vertex of P appears in more than one bag of $\mathcal{P}$. Since 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 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 in G_0 are contained in $\{K_\ell \cap K_{\ell+1} \mid \texttt{last}(i) \leqslant \ell < \texttt{first}(j)\}$. The set V_2 is defined to be the union of vertices in all induced v_1-v_p paths in G_0; according to Proposition 1.2, $V_2 \setminus \{v_1, v_p\} \subseteq \bigcup_{1<\ell<q} K_\ell$, and thus $V_2 \subseteq V_1$. 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.

Proposition 3.2. The vertex sets V_2 and A_0 are completely adjacent.

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 every bag of $\mathcal{P}$ containing x has smaller (resp., greater) index than every bag of $\mathcal{P}$ containing y. If an induced path of $G[V_2]$ consists of three or more vertices, then its ends 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 end of the path is said *to the left of the other*.



(a) A minimal solution (two deletions) that deletes a (dashed) edge incident to a simplicial vertex v.

(b) A minimal solution (3×3 deletions and $5+5$ additions) that adds (dotted) edges incident to a simplicial vertex v.

Figure 2: Possible modifications to simplicial vertices. (3 means a clique of 3 vertices).)

Lemma 3.3. Let C be a component of the subgraph induced by $V_1 \setminus V_2$. If C is nonadjacent to v_1 and v_p, then $N_{V_0}(C)$ induces a clique and there exists ℓ such that $1 < \ell < q$ and $N_{V_0}(C) \subseteq K_\ell$.

Proof. Since C is nonadjacent to v_1 and v_p, it is disjoint from K_1 and K_q. As a result, $N_{V_0}[C] \subseteq V_1$, and then $N_{V_0}(C) \subseteq V_2$. Recall that all vertices of V_2 appear in $\mathcal{P}$, and thus every clique of V_2 is a subset of some bag in $\mathcal{P}$; it suffices to show that $N_{V_0}(C)$ induces a clique. Suppose for contradiction that there is a pair of nonadjacent vertices $x, y \in N_{V_0}(C)$; without loss of generality, let x lie to the left of y. We can find an induced v_1-x path P_x through no vertices to the right of x, and an induced y-v_p path P_y through no vertices to the left of y. Let x' be the first vertex in P_x (counting from v_1) that is adjacent to C, and y' the last vertex in P_y (counting from y) that is adjacent to C. We can find an induced path $x'P'y'$ with all internal vertices from C. Note that x' either is x or lies to the left of x in P_x 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. $\square$

Such a component C is called a *branch* of P, and we say that C is *near to* some internal vertex v_i of 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 $N_{V_0}(C) \subseteq N[v_i]$; here note that since $G[V_1]$ is connected, $N_{V_0}(C) \neq \emptyset$. 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 Proposition 3.2 and Lemma 3.3, $N_{V_0}(C) \cup A_0$ is a clique, and thus a hole cannot both enter and leave C 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., $N(C)$ can be a clique even if it intersects $M \setminus A_M$. 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 may suggest 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 X. However, this is not true: as Figure 2 shows, if a solution removes some edges of $N(X)$, then the solution may also need to add/delete edges incident to X. As characterized by the following lemma, this is the only reason for touching X in the solution. In other words, a simplicial set X will only concern us after $N(X)$ has been changed; after all, X may not be simplicial any more. We say that a chordal editing set (V_-, E_-, E_+) *edits* a set $U \subset V(G)$ of vertices if either V_- contains a vertex of U or $E_- \cup E_+$

contains an edge incident to U. 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. Note that a branch is always simplicial in G_0 (by the definition of G_0 and Lemma 3.3), but it is 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, denoted by $[v_s, v_t]$, is a *segment* if v_s and v_t are the only junctions in it.

Because v_1 is adjacent to $M \setminus A_M$, both v_1 and v_2 are junctions of type (1); so are $\{v_{p-1}, v_p\}$. We point out that the four types are not exclusive, and one junction might be in more than one type. 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 path 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 might be a hole "local to v" (in the sense that it passes through only $M \cup A$ and vertices near to v), and its disposal might interfere with that of H. On the other hand, since there are 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 junctions and segments extends to all paths of $H - M$. Once H is given, we can find in polynomial time the sets A and V_0, and construct clique trees for each component of G_0; for each path P of $H - M$, we can find in polynomial time V_1 and V_2, and identify all its junctions. Since both ends of P are type (1) junctions (adjacent to $M \setminus A_M$), every vertex in $H - M$ 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 that 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, by Proposition 3.1, 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 $|M| \cdot (14k^2 + 88k + 82)$ segments, then we can either find a vertex that has to be in V_-, or return "NO."

Proof. Since there are at most $|M|$ paths in $H - M$, at least one of them contains more than $14k^2 + 88k + 82$ segments. Let P be such a path, then P has more than $14k^2 + 88k + 83$ junctions. Let us first attend to junctions of type (1) in P.

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

Proof. Suppose, for contradiction, that 15 or more vertices in P appear in bags 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 either consecutive in P or at the two ends of P (Prop. 3.1). Thus, we can always pick 6 consecutive vertices from X that are nonadjacent to 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 $u_2 \sim v_{i+5}$. Using Prop. 3.1 it is easy to verify that $u_2 \not\sim v_i, v_{i+1}, v_{i+2}$ and $u_1 \not\sim v_{i+3}, v_{i+4}, v_{i+5}$. Therefore, we can find an induced u_1-u_2 path with all internal vertices from $\{v_i, \ldots, v_{i+5}\}$. The length of this path is at least 3, and hence $u_1 \not\sim u_2$ (G_0 is chordal) and the path together with w makes a hole of length at most 9, contradicting the assumption that H is a shortest hole.

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 $v_\ell \notin X$ for every $i < \ell < j$. Note that neither of v_i and v_j can be adjacent to w, as otherwise v_{i+1} or v_{j-1} will also be a junction of type (1). There are two vertices $u_1, u_2 \in V_0 \cap N(w)$ such that $u_1 \sim v_i$ and $u_2 \sim v_j$. Neither of u_1 and u_2 can be adjacent to v_ℓ for $i < \ell < j$ (they are not junctions of type (1)); noting that $|X| \geqslant 15$, from Prop. 3.1 and the fact that G_0 is chordal we can conclude that $u_1 \not\sim u_2$, $u_1 \not\sim v_j$, and $u_2 \not\sim v_i$. Therefore, $w u_1 v_i \cdots v_j u_2 w$ is a hole. By assumption that $|X| \geqslant 15$, we have $j - i \leqslant |H| - 13$. Thus, we obtain a hole strictly shorter than H, a contradiction. ⌟

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 of type (1) or (2) witnessed by w. We order these vertices 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_{ℓ_1}, v_{r_1}), …, $(v_{\ell_{k+1}}, v_{r_{k+1}})$.

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 P is a proper subset of $N[v_{\ell_j}]$ (resp., $N[v_{r_j}]$). 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 $r_j - \ell_j > 2$. According to Proposition 3.1, v_{r_j} and v_{ℓ_j} have no common neighbors in V_0. Therefore, $N_{V_0}(C_{\ell_j})$ and $N_{V_0}(C_{r_j})$, which are nonempty subsets of $N_{V_0}[v_{\ell_j}]$ and $N_{V_0}[v_{r_j}]$ respectively, are disjoint. The neighborhoods of C_{ℓ_j} and C_{r_j} on the hole H are possibly empty, and this fact is irrelevant for the argument to follow. Since C_{ℓ_j} induces a connected subgraph and is adjacent to both w and $N_{V_0}[v_{\ell_j}]$, we can find an induced w-v_{ℓ_j+1} path P_{ℓ_j} with all internal vertices from $C_{\ell_j} \cup N_{V_0}[v_{\ell_j}]$. Likewise, we can obtain an induced w-v_{r_j-1} path P_{r_j} with all internal vertices from $C_{r_j-1} \cup N_{V_0}[v_{r_j}]$. These two paths P_{ℓ_j} and P_{r_j}, together with $v_{\ell_j+1} \ldots v_{r_j-1}$, make the hole H_j: we have $\ell_j + 1 < r_j - 1$; for each $\ell_j + 1 \leqslant s \leqslant r_j - 1$, $v_s \not\sim w$ (as there cannot be a junction of type (1) in between); 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 in the disjoint compression version of the problem, 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 to consider the set B of junctions that are only of type (3) or (4) but not of the first two types. The size of B is at least (noting $|M| \leqslant k + 1$)

$$(14k^2 + 88k + 83) - (5k + 74) \cdot |M| \geqslant 9k^2 + 9k + 9 = 9\,(k(k+1) + 1)\,.$$

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 $(9i+4)$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 $v_{b_{9i+4}}$ is of type (4): there is a pair of nonadjacent vertices $x \in N_{V_2}(v_{b_{9i+4}})$ and $w \in A$; according to Proposition 3.2, $w \in A_M$. In this case we can assume that x is adjacent to neither $v_{b_{9i+2}}$ nor $v_{b_{9i+6}}$; otherwise $xv_{b_{9i+2}}wv_{b_{9i+4}}x$ or $xv_{b_{9i+4}}wv_{b_{9i+6}}x$ is a 4-hole, which contradicts the fact that H is the shortest. In other words, x only appears in bags between $K_{\texttt{last}(b_{9i+2})+1}$ and $K_{\texttt{first}(b_{9i+6})-1}$. By the definition of V_2, there is an induced v_1-v_p path P' via x in $G[V_2]$. This path necessarily passes through both $N[v_{b_{9i+2}}]$ and $N[v_{b_{9i+6}}]$. We find the last neighbor x' of $v_{b_{9i+2}}$ on P' and the first neighbor x'' of $v_{b_{9i+6}}$ on P'; they must lie to the different sides of x. We can thus construct an induced $v_{b_{9i+2}}$-$v_{b_{9i+6}}$ path P_i through x in $G[V_2]$. Note that x' and x'' are adjacent to $v_{b_{9i+2}+1}$ and $v_{b_{9i+6}-1}$ respectively; by Proposition 3.1, this path does not visit $N[v_{b_{9i}}]$ or $N[v_{b_{9i+8}}]$. Starting from x, we traverse P_i to the left until the first vertex x_1 that is adjacent to w; the existence of such a vertex is ensured by the fact that $w \sim v_{b_{9i+2}}$. Similarly, we find the first neighbor x_2 of w in P_i to the right of x. Then the sub-path of P_i between x_1 and x_2, together with w, gives the hole H_i. By construction, no vertex of $H_i - w$ is adjacent to $v_{b_{9i}}$ or $v_{b_{9i+8}}$.

In the other case, $v_{b_{9i+4}}$ is of type (3): some branch C_i near to $v_{b_{9i+4}}$ 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.

- The subgraph induced by $N(C_i)$ is not a clique. Since $v_{b_{9i+4}}$ 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 $w \in A_M$ and $x \in N(C_i) \cap V_0$, which is necessarily in V_2. As C_i is near to $v_{b_{9i+4}}$, it must hold that $x \in N(v_{b_{9i+4}})$. However, then $v_{b_{9i+4}}$ is also of type (4), and it has already been discussed.

- Suppose now that $N(C_i)$ induces a clique and there is a hole H_i in $N[C_i]$. Since $v_{b_{9i+4}}$ is not of type (2), $N[C_i] \cap M \subseteq A_M$. Since $G - M$ is chordal, H_i must intersect 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 $v_{b_{9i+1}}$ or $v_{b_{9i+7}}$. 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 $v_{b_{9i+4}}$ 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 $v_{b_{9i+4}}$ and $v_{b_{9j+4}}$ is greater than 2 on the hole H).

In summary, 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 and it has size at most two (as A is a clique). If there is a $u \in A_0$ contained in at least $k+1$ holes of $\mathcal{H}$; we argue that we have to put u into V_-. Here note that it might be the case that these $k+1$ holes contain also a common vertex $w \in M$, and then the edge uw is also common to them. Suppose that we delete uw but not u. After the deletion of uw, each of these holes becomes a u-w path. Internal vertices of these paths are in $N[v_{b_{9i+4}}]$ or in branches near to $v_{b_{9i+4}}$ for some i, and thus internal vertices of different paths cannot be adjacent. Therefore, any two of these paths will together make a hole longer than $k+4$, not fixable by edge additions only. Therefore, at least k of these paths have to be broken, but we have only $k-1$ modifications left, so we still need to delete u. Otherwise, no vertex in A_0 is contained in more than k holes of $\mathcal{H}$. We can find $k+1$ distinct holes that intersect only in M. Some subset $\mathcal{H}'$ of $\mathcal{H}$ may share the same edge ww' with $w, w' \in M$, but a similar argument as above ensures us that at least $|\mathcal{H}'|$ edges need to be deleted from $\mathcal{H}'$. Therefore, we can return "NO." This concludes the proof. □

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 $(|V_S|, |E_S|)$. Again, by definition, V_S needs to be disjoint from $\{x, y\}$. 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 simple property 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 connected chordal graph F. The degenerate case where $E_S = \emptyset$ is well understood: V_S itself makes an x-y separator and can be easily found. Hence we may assume $E_S \neq \emptyset$. Let X and Y be the vertices in the components of $G - V_S - E_S$ that contain x and y respectively. We fix a clique tree $\mathcal{T}^F$ of F, and consider the subtree $\mathcal{T}^F(X)$ induced by bags intersecting X, and the subtree $\mathcal{T}^F(Y)$ induced by bags intersecting Y. By minimality, all edges of E_S are between X and Y, hence in $\mathcal{T}^F(X) \cap \mathcal{T}^F(Y)$, which is again a subtree of $\mathcal{T}^F$, and every bag in it intersects both X and Y. The following conclusions follow from the minimality of (V_S, E_S).

(1) All vertices in all bags of $\mathcal{T}^F(X) \cap \mathcal{T}^F(Y)$ are either in V_S or belong to $X \cup Y$; in the second case, every such vertex is incident to at least one edge of E_S.

(2) Every vertex in V_S is adjacent to both X and Y, thereby appearing in some bag of $\mathcal{T}^F(X) \cap \mathcal{T}^F(Y)$.

We remark that the remaining vertices of a bag not in the subtree $\mathcal{T}^F(X) \cap \mathcal{T}^F(Y)$ may belong to a different component of $G - V_S - E_S$ than X and Y.

Each bag K in the subtree $\mathcal{T}^F(X) \cap \mathcal{T}^F(Y)$ contains at most $|V_S| + |E_S| + 1$ vertices: the number of edges between any nontrivial partition of $K \setminus V_S$ is at least $|K \setminus V_S| - 1$, which has to be at most $|E_S|$. Likewise, the total number of vertices in all bags of this subtree is at most $|V_S| + 2|E_S|$: each vertex in these bags that are not in V_S is incident to at least one edge in E_S, and one edge is incident to two vertices. This inspires the following algorithm.

Lemma 4.2. Let x and y be a pair of nonadjacent vertices in a connected chordal graph F. For any pair (a, b) of nonnegative integers, we can find a mixed x-y separator of size at most (a, b) or assert its nonexistence in time $9^{a+b} \cdot |V(F)|^{O(1)}$.

Proof. We find first a minimum vertex x-y separator S; if its size is at most a, then $(S, \emptyset)$ will be the mixed x-y separator. Henceforth we may assume that $E_S \neq \emptyset$ in any mixed x-y separator satisfying $|V_S| \leqslant a$. By previous discussion, to find such a mixed separator, it suffices to find the subtree $\mathcal{T}^F(X) \cap \mathcal{T}^F(Y)$ and a tri-partition of all vertices in all bags of this subtree, where X and Y are the components of $G - V_S - E_S$ containing x and y respectively.

To begin with, we guess a bag K of size at most $a + b + 1$ vertices, and generate in $3^{|K|}$ time all tri-partitions of K. If a mixed separator of size at most (a, b) exists, then in some branch we will guess a bag K in $\mathcal{T}^F(X) \cap \mathcal{T}^F(Y)$ along with its tri-partition whose three parts are in V_S, X, and Y respectively. We grow the bag K to the subtree $\mathcal{T}^F(X) \cap \mathcal{T}^F(Y)$ by considering its neighboring bags one by one. For such a bag K', we consider $K \cap K'$, of which all vertices have already been decided. If they are all in V_S, then their deletion separates the rest of K' (as well as all vertices in the subtree containing K' in $\mathcal{T}^F - K$) from both x and y, and thus they will not further concern us. A similar situation is when $(K \cap K') \setminus V_S$ a subset of X or Y, then the rest of K' will be in X or

Y accordingly: all paths connecting them to the other side go through $(K \cap K') \setminus V_S$. Otherwise, $K \cap K'$ intersects both X and Y, and then K' must be in $\mathcal{T}^F(X) \cap \mathcal{T}^F(Y)$. Let a' the number of vertices that have been decided to be in V_S, and let b' be the number of edges between vertices that have been decided in X and Y. If $K' \setminus K$ contains more than $(a - a') + (b - b') + 1$ vertices, then we terminate this branch; otherwise we guess a tri-partition for it and proceed to the next bag.

Extending this way, in the whole process all bags that have been partitioned form a subtree. Either all partitions have been terminated and we can conclude that there is no mixed x-y separator of size at most (a, b), or we obtain a mixed separator when it cannot be further extended, i.e., for every bag K' adjacent to some bag K of this subtree, $(K' \cap K) \setminus V_S$ are fully contained in either X or Y. The execution of this algorithm can be viewed as traversing a bounded search tree, and to bound the number of its leaves, we use $a + b$ as the measure. There are at most $|V(F)|$ bags in $\mathcal{T}^F$, which is the number of child nodes of the root node in the search tree. For each new bag with p undecided vertices, we have at most $3^p = 9^{p/2}$ tri-partitions to consider. For each of these tri-partitions, we have a sub-instance, where the measure decreases by at least $p - 1$. Noting that $p/2 < p - 1$, the total number of leaves is at most $9^{a+b} \cdot |V(F)|$, and the running time of the algorithm is $9^{a+b} \cdot |V(F)|^{O(1)}$. This completes the proof. □

The definition of mixed separator can be easily generalized to two disjoint vertex sets each inducing a connected subgraph—we may simply contract each set into a single vertex (after which the graph remains chordal) 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 F be a chordal graph, and let X and Y be a pair of disjoint and nonadjacent sets of vertices in F such that both $F[X]$ and $F[Y]$ are connected. For any nonnegative integer $a \leqslant k_1$, in time $9^{k_1+k_2} \cdot |V(F)|^{O(1)}$ 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).

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. On the other hand, we are not aware of any proof of its NP-hardness on chordal graphs, and its complexity is still open.

5 Proof of Theorem 2.1

We are now ready to put everything together and finish the algorithm for CHORDAL EDITING COMPRESSION. We say that a chordal editing set is minimum if there exists no chordal editing set with a smaller size.

Proof of Theorem 2.1. We start from finding a shortest hole H. If $|H| \leqslant k + 3$, then we try one of the $|V(H) \setminus M|$ vertex deletions, $|H|$ edge deletions, and $O(|H|^2)$ edge insertions that affect H. Clearly, at least one of these operations is necessary, and each of them makes a new instance that has strictly smaller parameters. Hence we may assume $|H| \geqslant k + 4 > k_3 + 3$. With H fixed, we have A (common neighbors of H) and V_0 (i.e., $V(G) \setminus (M \cap A)$) defined; we further build clique trees for all components in $G_0 := G[V_0]$ and find all segments of H. If there are more than $|M| \cdot (14k^2 + 88k + 74)$ segments, then by Theorem 3.6, we have either found a vertex that must be in V_- in any valid solution, leading to a new instance with smaller parameters, or been ready to return "NO." Henceforth, we may assume that H contains $O(k^3)$ segments, which also means that it has $O(k^3)$ junctions.

Let us fix a hypothetical (V_-^*, E_-^*, E_+^*) minimum chordal editing set of G of size no more than (k_1, k_2, k_3). There are three options for breaking H by this set. In the first case, V_-^* contains some junction, or E_-^* contains some edge of H that is incident to M. 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 an 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 hole H) from a junction; or (2) an edge whose both endvertices are at distance at most d (on the hole H) 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. The vertex or edge we need to delete from H must then belong to some segment $[v_s, v_t]$ with $t - s > 2d$; in particular, it is in the sub-path $v_{s'} \cdots v_{t'}$, where $s' := s + d$ and $t' := t - d$. This is the third case and our main concern. 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, and $\mathcal{T}$ is a clique tree for the component of the chordal subgraph G_0 that contains P. 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. Suppose that for every segment $[v_s, v_t]$ with $t - s > 2d$, vertices $v_{s'}$ and $v_{t'}$ remain 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 this path $v_s \cdots v_{s'}$ and $v_{t'} \cdots v_t$, (which are assumed to be not impacted by the deletion of V_-^* and E_-^*,) 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 $G_{[s,t]}$. By the definition of $G_{[s,t]}$, the path $P'_{[s,t]}$ must be disjoint from A. The distance between v_s and v_t in $G_{[s,t]}$ must be $t - s > 2d$; otherwise we have a hole shorter than H (whether some internal vertex of $P'_{[s,t]}$ is adjacent to $H \setminus [v_s, v_t]$ or not), which is impossible. Therefore, the length of $P'_{[s,t]}$ is larger than $2d > 2k_3 + 4$.

We have assumed that any segment of length at most 2d remains intact in $G - V_-^* - E_-^*$. Therefore, we have now 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 incident to M, we get a closed walk C, which is disjoint from A. To show that C is a hole, it suffices to verify that any vertex $v \in V(C) \setminus V(H)$ is nonadjacent to other vertices of C different from its two neighbors in C. By construction, v belongs to some path $P_{[s',t']}$ given above, which is either in $N_{V_2}[v_{s'}], \ldots, N_{V_2}[v_{t'}]$, or in some branch near to $v_{s'}, \ldots, v_{t'}$; since none of $v_{s'}, \ldots, v_{t'}$ is a junction of type (1) or (2), it follows that v is not adjacent to $M \setminus A_M$, and thus $N_V(v) \subseteq V_0$. Suppose that besides its two neighbors in C, the vertex v is adjacent to another vertex u in C, then u cannot be in $P'_{[s,t]}$ and cannot be in H. In other words, u is in a branch near to some vertex v_j that is at least 2d far away from $v_{s'}, \ldots, v_{t'}$, which is impossible as G_0 is chordal. Hence, C must be a hole of $G - V_-^* - E_-^*$ of length larger than $2k_3 + 4$. It cannot be made chordal by the addition of the at most k_3 edges of E_+^*, and 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 in $G_{[s,t]}$. The resulting graph obtained by deleting this mixed separator from $G_{[s,t]}$ is characterized by the following claim.

Claim 4. Let $G' = G - V_S - E_S$, where (V_S, E_S) is an inclusion-wise minimal mixed $\{v_s, \ldots, v_{s'}\}$-$\{v_{t'}, \ldots, v_t\}$ separator in $G_{[s,t]}$. For any i with $s + 2 \leqslant i \leqslant s' - 2$, the component X of $G' - (K_{\mathtt{last}(i)} \cup A)$ containing $v_{s'}$ is simplicial in G'.

Proof. We argue first that X is the same as the component of $G_{[s,t]} - (K_{\mathtt{last}(i)} \cup V_S) - E_S$ containing $v_{s'}$. Let X' be the component; note that it fully contains the path $v_{i+2} \cdots v_{s'}$. Since $G_{[s,t]}$ is a subgraph of $G - A$, it follows that $X' \subseteq X$. Note that no $v_{s'}$-v_t path in $G_{[s,t]}$ can be shorter than d (otherwise there is a hole shorter than H). As a result, X' cannot contain a neighbor of v_t: since X' is connected, a $v_{s'}$-v_t path will imply that $|E_S| > k$. Therefore, X' has no neighbor in $K_{\mathtt{last}(s)} \cup K_{\mathtt{first}(t)}$. On the other hand, a vertex in X' is either in $N[v_{i+1}], \ldots, N[v_{t-1}]$ or a branch near to $v_{i+1}, \ldots, v_{t-1}$, and hence cannot be adjacent to $M \setminus A_M$. Suppose for contradiction that $X' \neq X$, then we can find a neighbor of X' in $M \setminus A_M$ or $K_{\mathtt{last}(s)} \cup K_{\mathtt{first}(t)}$, but we have seen that it is not possible. Hence, $X \subseteq V_0$, and since $K_{\mathtt{last}(i)}$ and (V_S, E_S) have been deleted, it can be further inferred $X \subseteq V_1$. Then $N_{G'}(X) \subseteq K_{\mathtt{last}(i)} \cup A$. Since $v_{i+1} \in K_{\mathtt{last}(i)}$ and is not a junction of type (4), $N_{G'}(X)$ must be a clique. 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'.

The subgraph induced by $N_{G'}[X] \cap V_2$ is chordal, and since every vertex in it is adjacent to some vertex in $v_{i+1}, \ldots, v_{t'-1}$, which is not a junction of type (4), they are completely adjacent to A. All other vertices in $N_{G'}[X]$ (not in $V_2 \cup A$) are in some branch near to $v_{i+1}, \ldots, v_{t'-1}$. Since these vertices are not junctions of type (3), these branches are all simplicial in G. Let C be such a branch; according to Lemma 3.3, $N_{V_2}(C)$ is a clique, and $N(C) \setminus N_{V_2}(C)$ are in A and hence $N(C)$ is a clique. 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 $t' + 2 \leqslant i \leqslant t - 2$, the component X of $G' - (K_{\mathtt{first}(i)} \cup A)$ that contains $v_{t'}$ is simplicial in G'. Let (V_S^*, E_S^*), where $V_S^* \subseteq V_-^*$ and $E_S^* \subseteq E_-^*$, be an inclusion-wise minimal mixed $\{v_s, \ldots, v_{s'}\}$-$\{v_{t'}, \ldots, v_t\}$ separator in $G_{[s,t]}$. 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 the desired 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 running time is thus $O(k)^{4k} \cdot n^{O(1)} = 2^{O(k \log k)} \cdot n^{O(1)}$. This completes the proof. $\square$

6 Concluding remarks

We would like to draw attention to the similarity between CHORDAL VERTEX 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 their 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 and probably better way to 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 let S_ι have the minimum cardinality among $\{S_i : \texttt{last}(s) \leqslant i < \texttt{first}(t)\}$. If there exists S_ℓ such that S_ℓ is disjoint from $K_{\texttt{last}(s)} \cup K_{\texttt{first}(t)}$ 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. Simply deleting vertices of degrees one or two already suffices to yield a linear-vertex kernel for the DISJOINT FEEDBACK VERTEX SET problem, the compression variant of FEEDBACK VERTEX SET [7]. However, for our problem it does not seem to be the case, and to furnish a polynomial kernel for even the compression variant of the CHORDAL VERTEX DELETION problem, we might need more than Reductions 1 and 2. We leave the existence of polynomial kernels for CHORDAL VERTEX DELETION and its compression variant as an open problem.

We have presented the first FPT algorithm for the general modification problem to a graph class that has an infinite number of obstructions. Following this work, the first author [6] showed that the unit interval editing problem is FPT as well. 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 would be the interval graphs.

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

Acknowledgment. We are grateful to the anonymous referees for their careful reading and helpful suggestions, and in particular for pointing out flaws in a preliminary version and suggesting a better proof for Lemma 4.2.

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. doi:10.1137/0215075.

[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] Claude Berge. Motivations and history of some of my conjectures. *Discrete Mathematics*, 165-166:61–70, 1997. doi:10.1016/S0012-365X(96)00161-6.

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

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

[6] Yixin Cao. Unit interval editing is fixed-parameter tractable. In *Automata, Languages and Programming (ICALP), to appear*, 2015. Full version available at arXiv:1504.04470.

[7] Yixin Cao, Jianer Chen, and Yang Liu. On feedback vertex set: New measure and new structures. *Algorithmica*, 2014. In press. doi:10.1007/s00453-014-9904-6.

[8] P.M. Dearing, D.R. Shier, and D.D. Warner. Maximal chordal subgraphs. *Discrete Applied Mathematics*, 20(3):181–190, 1988. doi:10.1016/0166-218X(88)90075-3.

[9] Gabriel A. Dirac. On rigid circuit graphs. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 25(1):71–76, 1961. doi:10.1007/BF02992776.

[10] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Undegraduate texts in computer science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

[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. doi:10.1137/S0097539796303044.

[14] Mukkai S. Krishnamoorthy and Narsingh Deo. Node-deletion NP-complete problems. *SIAM Journal on Computing*, 8(4):619–625, 1979. doi:10.1137/0208049.

[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. doi:10.1137/0910070.

[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. doi:10.1016/0022-0000(80)90060-4.

[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. doi:10.1016/j.tcs.2005.10.008.

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

[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. doi:10.1145/2500119.

[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 WG 1999. doi:10.1016/S0166-218X(00)00391-7.

[22] Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004. doi:10.1016/j.orl.2003.10.009.

[23] Donald J. Rose. Triangulated graphs and the elimination process. *Journal of Mathematical Analysis and Applications*, 32(3):597–609, 1970. doi:10.1016/0022-247X(70)90282-9.

[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] Jue Xue. Edge-maximal triangulated subgraphs and heuristics for the maximum clique problem. *Networks*, 24(2):109–120, 1994. doi:10.1002/net.3230240208.

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