A parameterized algorithm for CHORDAL SANDWICH*

Pinar Heggernes† Federico Mancini† Jesper Nederlof†
Yngve Villanger†

Abstract

Given an arbitrary graph $G = (V, E)$ and an additional set of admissible edges F, the CHORDAL SANDWICH problem asks whether there exists a chordal graph $(V, E \cup F')$ such that $F' \subseteq F$. This problem arises from perfect phylogeny in evolution and from sparse matrix computations in numerical analysis, and it generalizes the widely studied problems of completions and deletions of arbitrary graphs into chordal graphs. As many related problems, CHORDAL SANDWICH is NP-complete. In this paper we show that the problem becomes tractable when parameterized with a suitable natural measure on the set of admissible edges F. In particular, we give an algorithm with running time $\mathcal{O}(2^k n^5)$ to solve this problem, where k is the size of a minimum vertex cover of the graph (V, F). Hence we show that the problem is fixed parameter tractable when parameterized by k. Note that the parameter does not assume any restriction on the input graph, and it concerns only the additional edge set F.

1 Introduction

After arising from practical problems and appearing in various papers under different names, graph sandwich problems were formalized into a general framework by Golumbic et al. [16]. For a given graph class Π, the Π-SANDWICH problem takes as input a pair (G, F), where $G = (V, E)$ is a graph and $F \subseteq (V \times V) \setminus E$ is a set of additional edges, and asks whether there is a subset F' of F such that the graph $(V, E \cup F')$ belongs to the class Π. Taking $H = (V, E \cup F)$, we say informally that we look for a graph belonging to class Π *sandwiched between* G *and* H. The Π-SANDWICH problem is NP-complete when Π is one of the following graph classes: chordal, interval, circle, circular arc, comparability, permutation, strongly chordal, and chordal bipartite [1, 16, 21], whereas it can be solved in polynomial time when Π is the class of split graphs, threshold graphs or cographs [16]. The Π-SANDWICH

*This work is supported by the Research Council of Norway.

†Department of Informatics, University of Bergen, N-5020 Bergen, Norway. Emails: {pinar.heggernes, federico.mancini, jesper.nederlof, yngve.villanger}@ii.uib.no

problem has also been studied with respect to other graph classes and properties Π [8, 18, 22, 31], and on hypergraphs [17].

Chordal graphs are the class of graphs that do not contain any chordless cycle of length greater than 3 as induced subgraphs. In general, chordal graphs constitute one of the most studied graph classes, since problems from many different fields can be interpreted as chordal graph problems [15]. The Chordal Sandwich problem is one of the sandwich problems with most applications. It arises in particular in sparse matrix computations [16] and in perfect phylogeny since it has the problem of triangulating colored graphs as a special case [1, 2]. It can also be seen as a generalization of the problems of adding or deleting edges in a minimum or minimal way in an arbitrary input graph to obtain a chordal graph, which have attracted considerable attention [13, 14, 19, 20, 23, 24, 26, 27]. The NP-completeness of the problem follows from the results of several papers [1, 6, 32]. After being proved NP-complete, the only tractability result on Chordal Sandwich on arbitrary inputs is an algorithm by Lokshtanov [25]. Combining this with a result of [14] gives an algorithm with running time $\mathcal{O}^*(1.7549^n)$, where the $\mathcal{O}^*$-notation suppresses the terms that are polynomial in the size of the input.

We attack the Chordal Sandwich problem from a parameterized perspective. A graph problem is called *Fixed Parameter Tractable*, or FPT for short, if we can identify a parameter k, typically an integer, such that there is an algorithm solving the problem with running time $f(k) \cdot n^{O(1)}$, where n is the number of vertices in G, and f is a computable function depending only on the parameter k.

The choice of a suitable parameter is very important in the design of parameterized algorithms. The most notable example is *treewidth*. Many hard graph problems become FPT when parameterized by the treewidth of the input graph [10, 12, 28], but our problem is harder than so. In fact, when parameterized by the treewidth of G, it follows from the results of Bodlaender et al. [2] that the Chordal Sandwich problem is $W[t]$-hard for every integer $t \geq 1$. Regarding the applications of Chordal Sandwich, it is reasonable to assume that one has some bound or control on the additional set of admissible edges F, since these represent the unreliable data. However, parameterizing by $|F|$ is not really interesting, as the problem then becomes trivially FPT, since we can try all possible ways of adding subsets of F to G. Another parameterization could be $|F'|$, the number of edges we end up adding to G to make it chordal within the restrictions given by $F' \subseteq F$. But also this case is easily FPT, following the fixed parameter tractability of the Minimum Fill-In problem [23, 24, 7, 3], since one can simply discard the solutions that do not fit into F while computing all chordal completions that add at most $|F'|$ edges. For completeness, it should be mentioned that when we parameterize by the treewidth of $H = (V,\ E \cup F)$, the problem is again easily shown to be FPT, as also noted in [25].

We study the following parameterization:

k-Chordal Sandwich

Instance:	A graph $G = (V, E)$ and a set $F \subseteq (V \times V) \setminus E$.
Parameter:	An integer k such that there exists a vertex cover of the graph (V, F) of size at most k.
Question:	Is there a chordal graph $M = (V, E \cup F')$ where $F' \subseteq F$?

Notice that deciding whether a given graph has a vertex cover of size at most k is FPT [10]. Based on the above discussion, our parameter k seems natural for this problem; it is never larger than $|F|$ and in most cases it is much smaller, thus as parameter it gives less restriction than $|F|$ on the admissible edges. Furthermore it gives no restriction on the input graph G.

In this paper we give an algorithm with running time $\mathcal{O}(2^k n^5)$ that solves the k-Chordal Sandwich problem. Consequently, we show that k-Chordal Sandwich is FPT. We achieve our results by bounding the number of combinatorial objects that need to be handled. In particular, we show that the number of minimal separators and potential maximal cliques that can be useful to achieve a solution is bounded by $\mathcal{O}(2^k n^2)$. Combining this with a similar approach to that of [13] in such a way that we only consider potential maximal cliques that can lead to a solution, gives the running time $\mathcal{O}(2^k n^5)$.

It should be mentioned that in general, the number of minimal separators and the number of potential maximal cliques of a graph can be as large as $\mathcal{O}^*(1.7549^n)$. The construction is simple, and can for instance be found in [13]. It is thus interesting to see that the set F of admissible edges implies a strong enough restriction to be able to bound the number of such objects that are useful in a solution to a function that is only exponential in the size of the minimum vertex cover of F. To our knowledge this technique has not been used to determine the complexity of a problem before.

2 Preliminaries

All graphs in this work are undirected and simple. A graph is denoted by $G = (V, E)$, with vertex set $V = V(G)$ and edge set $E = E(G)$. We let $n = |V|$ and $m = |E|$. A set R of vertices is a *vertex cover* of a graph if every edge of the graph has an endpoint in R. For a vertex subset $S \subseteq V$, the subgraph of G induced by S is denoted by $G[S]$. If a subset $D \subseteq E$ of the edges is given, we will denote by $G[D]$ the subgraph of G induced by the set of endpoints of all edges in D. We denote the neighborhood of S in G by $N_G(S) = \{v \in (V \setminus S) \mid \exists w \in S : \{v, w\} \in E\}$. We write $N_G(v) = N_G(\{v\})$ for a single vertex v, and $N_G[S] = N_G(S) \cup S$. Subscripts are omitted when not necessary. A vertex set is a *clique* if every pair of vertices in it are adjacent. A clique is *maximal* in G if it is not a proper subset of another clique in G.

A vertex subset $S \subseteq V$ is a *separator* in G if $G[V \setminus S]$ has at least two connected components. A connected component C is a *full* component associated to S if

$N(C) = S$. A vertex set S is a *uv-separator* for G if u and v are in different connected components of $G[V \setminus S]$, and a *minimal uv-separator* for G if no proper subset of S is a uv-separator. In general, a vertex subset S is a *minimal separator* of G if there exist u and v in V such that S is a minimal uv-separator. A set of vertices that is both a clique and a separator is called a *clique separator.*

Proposition 1 (Folklore)**.** *A set S of vertices in a graph G is a minimal uv-separator if and only if u and v are in different full components associated to S. In particular, S is a minimal separator if and only if there are at least two distinct full components associated to S.*

A *chord* in a cycle (path) is an edge between two non-consecutive vertices of the cycle (path). A graph is *chordal* if every cycle on four or more vertices has a chord. A graph $M = (V, E \cup F')$ is a *triangulation* or *chordal completion* of a graph $G = (V, E)$ if M is chordal. The edges in $F' \setminus E$ are called *fill edges.* M is a *minimal triangulation* of G if there is no triangulation $M' = (V, F'')$ of G with $F'' \subset F'$. It is known that if a triangulation $M = (V, E \cup F')$ is not minimal, then there exists an edge $\{u, v\} \in F'$ such that $M' = (V, E \cup (F' \setminus \{\{u, v\}\})$ is chordal [30]. As a consequence, this implies that there exists a chordal graph $M = (V, E \cup F')$ where $F' \subseteq F$ if and only if there exists a minimal triangulation $M' = (V, E \cup F'')$ of G where $F'' \subseteq F$. Thus, the chordal sandwich problem is equivalent to asking if there exists a minimal triangulation $M' = (V, E \cup F'')$ of G where $F'' \subseteq F$. Two separators S_1 and S_2 are said to be *crossing* if S_1 is a uv-separator for two vertices $u, v \in S_2$, in which case S_2 is an xy-separator for two vertices x, y in S_1 [29]. The following characterization of minimal triangulations related to minimal separators is important for understanding our results.

Theorem 2 ([29])**.** *Given an arbitrary graph G, a chordal graph M is a minimal triangulation of G if and only if M is the result of making a maximal set of pairwise non-crossing minimal separators into cliques (by adding necessary edges to G to achieve this).*

A *potential maximal clique* of a graph G is defined as a maximal clique in a minimal triangulation of G. Fortunately it is easy to check if a vertex set is a maximal clique of some minimal triangulation.

Theorem 3 ([5])**.** *Let $\Omega \subseteq V$ be a set of vertices of the graph $G = (V, E)$, and let $C_1, \ldots, C_p$ be the set of connected components of $G[V \setminus \Omega]$. Then Ω is a potential maximal clique if and only if :*

1. *$G[V \setminus \Omega]$ has no full component associated to Ω, and*
2. *for every pair of vertices $x, y \in \Omega$, $\{x, y\} \in E$ or $x, y \in N(C_i)$ for some $i \in \{1, \ldots, p\}$.*

Proposition 4. *For a potential maximal clique Ω of the graph G, let X be the connected component containing vertex $x \in \Omega$ in the graph $G[V \setminus (\Omega \setminus \{x\})]$. Then $G[X]$ is connected, and $\Omega = N(X) \cup \{x\}$.*

Proof. This follows more or less directly from Theorem 3. By definition of X, $N(X) \subset \Omega$. Every connected component C of $G[V \setminus \Omega]$ where $x \in N(C)$, $C \subset X$, and by Theorem 3 vertex $\{x, y\} \in E$ or $x, y \in N(C)$ for some connected component C of $G[V \setminus \Omega]$, so $N(X) = \Omega \setminus \{x\}$. □

A *tree decomposition* of a graph $G = (V, E)$ is a tree T whose nodes correspond to subsets of V, called *bags*, that satisfies the following: every vertex of V appears in a bag; for all $\{u, v\} \in E$, there is a bag where u and v appear together; for any vertex $u \in V$, the nodes of T corresponding to the bags that contain u induce a connected subtree of T. (We will simply use bags to denote both bags and nodes.) It follows that for two bags X and Y of a tree decomposition T, $X \cap Y$ is contained in every bag on the unique path between X and Y in T. A *clique tree* is a special kind of tree decomposition with a bijection between the nodes of the tree and the maximal cliques of G. A graph G has a clique tree if and only if G is chordal [6]. Minimal triangulations are chordal graphs, and a potential maximal cliques is defined as maximal clique of a minimal triangulation. Thus, we can rephrase the Chordal Sandwich problem once again: There exists a chordal graph sandwiched between G and H if and only if a tree decomposition of G can be obtained by only using bags that are potential maximal cliques of G.

If we view each edge of a clique tree as the intersection of its endpoint maximal cliques, every edge of a clique tree of G corresponds to a minimal separator of G, and every minimal separator of G appears as an edge in every clique tree of G [6]. This also implies that every minimal separator is a clique [9]. It is also important to note that chordal graphs have at most n maximal cliques [9], and hence at most $n-1$ minimal separators. Chordal graphs can be recognized and clique trees can be computed in linear time [15]. Induced subgraphs of chordal graphs are also chordal.

3 Bounding the number of useful minimal separators and potential maximal cliques

Assume that we are given a graph $G = (V, E)$, and an additional graph $H = (V, E \cup F)$ where $E \cap F = \emptyset$. Let R be a minimum vertex cover of (V, F), with $k = |R|$. We want to answer the question whether there exists an edge set $F' \subseteq F$ such that $(V, E \cup F')$ is chordal. Let us define $U = V \setminus R$; hence F contains no edge with both endpoints in U. We can therefore observe that $G[U]$ has to be chordal, since any induced cycle in $G[U]$ cannot be triangulated by adding edges from F. By the results of [13], we can "search through" all minimal triangulations of a graph if we have a list of all its minimal separators and potential maximal cliques. This search is linear in the number of minimal separators and potential maximal cliques but requires an additional factor of $\mathcal{O}(n^3)$ time. In general graphs the number of minimal separators and potential maximal cliques might be exponential in the number of vertices; $\mathcal{O}^*(3^{n/3})$ is the best known lower bound, and $\mathcal{O}^*(1.6181^n)$ and $\mathcal{O}^*(1.7549^n)$ are the best known upper bounds for respectively the number of minimal separators and the number of potential maximal cliques [14]. Here, we will show that the restrictions given by F imply enough structure to bound the number of useful minimal separators and potential maximal cliques by a smaller function. First we will turn our attention towards listing minimal separators. Instead of finding the minimal separator directly we will find one of the two full components associated to the minimal separator, and retrieve the minimal separator from the neighborhood of this vertex set.

Due to Theorem 2, we can only use minimal separators of G that are cliques in H. Every minimal separator S of G, which is a clique in H, has the property that $G[S \cap U]$ is a clique, since we explained that no edge of F has two endpoints in U. Thus, we are interested in listing only such minimal separators.

Lemma 5. *The number of minimal separators S in G such that $G[U \cap S]$ is a clique is at most $2^k \cdot n^2$, and these can be listed in $\mathcal{O}(2^k \cdot n^4)$ time.*

Proof. By Proposition 1 we can uniquely define the vertex set of a minimal separator by obtaining one of the two full components associated to the minimal separator. The minimal separator can then be retrieved from the neighborhood of this vertex set. Hence, to proof the Lemma, it suffices the to list a set of full components such that for each minimal separator S, at least one full component associated to S is listed. Let S be a minimal separator such that $G[U \cap S]$ is a clique, and let X be a maximal clique of $G[U]$, such that $S \cap U \subseteq X$. Vertex set $X \setminus S$ induces a clique, and is thus contained in the same connected component of $G[V \setminus S]$. A result of this is that one of the two full connected components associated to S has an empty intersection with X. Let C be the vertex set of this full component, let $R_C = R \cap C$, and let x be a vertex contained in C. The connected component C of S will now be the connected component of $G[V \setminus (X \cup (R \setminus R_C))]$ which contains the vertex x.

Now the set of minimal separators can be listed in the following way. For each triple X, R_C, x where X is a maximal clique of $G[U]$, R_C is a subset of R, and $x \in V$ we define Z as $N(C')$ for the connected component C' of $G[V \setminus (X \cup (R \setminus R_C))]$ containing vertex x. Vertex set Z can be obtained in $\mathcal{O}(n^2)$ time. In the same time we can check if $G[V \setminus Z]$ contains two full components associated to Z, and thus verifying if Z is a minimal separator. If Z is a minimal separator we add it to the output list. The induced graph $G[U]$ is chordal, and thus $G[U]$ contains at most $|U| \leq n$ maximal cliques. Moreover, there are at most 2^k choices for R_C and at most n choices for x, hence this procedure requires $\mathcal{O}(2^k \cdot n^4)$ time. □

In a similar way we will now bound the number of potential maximal cliques Ω, where $G[U \cap \Omega]$ is a complete graph. Analogous to the explanation on minimal separators above, a potential maximal clique Ω can be a maximal clique in a solution sandwich graph that we are looking for, only if Ω is a clique in H. Since no edges can be added to G with both endpoints in U, this is equivalent to saying that $G[U \cap \Omega]$ is a complete graph.

For a vertex $x \in \Omega$, define Z to be the connected component of $G[V \setminus (\Omega \setminus \{x\})]$ that contains x. By Proposition 4 $\Omega = N(Z) \cup \{x\}$. The pair (Z, x) will be referred to as a *vertex representation* of Ω. For vertex set $Y = N(Z) \cup \{x\}$ we can use the properties of Theorem 3 as described in [5] to verify that there are no full components associated to Y, and that for every pair of vertices $x, y \in Y$, $\{x, y\} \in E$ or $x, y \in N(C)$ for some connected component of $G[V \setminus Y]$ in time $\mathcal{O}(nm)$.

For a minimal separator it was enough to know one of the full components associated to the minimal separator. In a similar way we will show that it is enough to find one of the $|\Omega|$ vertex representations of Ω.

Lemma 6. *The number of potential maximal cliques Ω in G such that $G[U \cap \Omega]$ is a clique is at most $2^k \cdot n^2$, and these can be listed in $\mathcal{O}(2^k \cdot n^5)$ time.*

Proof. Let X be a maximal clique of $G[U]$ such that $\Omega \cap U \subseteq X$. In the same was as in the proof for minimal separators, we can argue that $X \setminus \Omega$ is a clique in G, and thus $X \setminus \Omega$ intersects only one connected component C of $G[V \setminus \Omega]$. By Theorem 3, $N(C) \subset \Omega$, so let x be a vertex in $\Omega \setminus N(C)$. Vertex representation (Z, x) of Ω will now have an empty intersection with X. Let R_Z be the intersection $Z \cap R$, and notice that the connected component of $G[V \setminus (((R \setminus R_Z) \cup X) \setminus \{x\})]$ containing x is exactly Z, where (Z, x) is the vertex representation for Ω.

The set of potential maximal cliques are listed in the same was as we listed minimal separators. For each triple X, R_Z, x where X is a maximal clique of $G[U]$, R_Z is a subset of R, and $x \in V$ we define Z as the connected component of $G[V \setminus ((X \cup (R \setminus R_C)) \setminus \{x\})]$ containing vertex x. Vertex set Y is defined as $N(Z) \cup \{x\}$, and can be obtained in $\mathcal{O}(n^2)$ time. Use now the algorithm defined in [5] to verify in time $\mathcal{O}(n^3)$ if the vertex set Y is a potential maximal clique, and add it to the output list if this is the case. In total this require at most $\mathcal{O}(2^k \cdot n^5)$ time. □

The bounds for the number of minimal separators and potential maximal cliques of the type bounded in Lemmas 5 and 6 are optimal up to a polynomial factor of n. This follows from the following graph G: Let U be vertex set $u_0, u_1, \dots, u_k$ and let R be vertex set $v_0, v_1, \dots, v_k$. The graph $G[U]$ is a clique, $\{v_0, v_i\} \in E$ for every $i \in \{1, 2, \dots, k\}$, and $\{v_i, u_i\} \in E$ for every $i \in \{1, 2, \dots, k\}$. Graph H is a complete graph. Let X be any of the 2^k subsets of $R \setminus \{v_0\}$. Then $N(X \cup \{v_0\})$ is a minimal v_0, u_0 separator. It is not hard to verify that each such minimal separator also defines a unique potential maximal clique.

4 A parameterized algorithm for k-Chordal Sandwich

We will now use the list of minimal separators and potential maximal cliques to search for a minimal triangulations of G that can be created by adding edges from F. Our approach is similar to those of [13] and [25]. However, instead of searching through all minimal triangulations, we need only to search through minimal triangulations that fit into H, and hence we need only to consider the special types of minimal separators and potential maximal cliques that we studied in the previous section.

For a minimal separator S of G, and a connected component C of $G[V \setminus S]$ we define (C, S) as a *block*, and the block is refereed to as *full* if $N(C) = S$. Let $\alpha(C, S)$ be a function on the block (C, S), defined as follows:

$$\alpha(C,S) = \begin{cases} 1, & \text{if } H[S] \text{ is a complete graph, and there exists a chordal graph} \\ & \text{sandwiched between } G'[C \cup S] \text{ and } H[C \cup S] \text{, where } G' \text{ is obtained} \\ & \text{from } G \text{ by completing } S \text{ into a clique,} \\ 0, & \text{otherwise.} \end{cases}$$

The α function will now tell us whether or not the block (C, S) can be used in a partial solution for the chordal sandwich problem. In addition to this we also need a way to describe a "big" block from several smaller blocks. This is obtained by using potential maximal cliques. For a full block (C, S) and a potential maximal clique Ω where $S \subset \Omega \subseteq C \cup S$, we define (C, S, Ω) as a *good triple*. In addition to this we define function $\beta(C, S, \Omega)$ as follows:

$$\beta(C,S,\Omega) = \begin{cases} 1, & \text{if } H[\Omega] \text{ is a complete graph, and there exists a chordal graph} \\ & \text{sandwiched between } G'[C \cup S] \text{ and } H[C \cup S] \text{, where } G' \text{is} \\ & \text{obtained from } G \text{ by completing } \Omega \text{ into a clique,} \\ 0, & \text{otherwise.} \end{cases}$$

Remember that there exists a chordal graph $(V, E \cup F')$ sandwiched between G and H, if and only if there exists a minimal triangulation $M = (V, E \cup F'')$ sandwiched between G and H. This implies that there exists a minimal separator S of G, such that $H[S]$ is a clique and $\alpha(C, N(C)) = 1$ for every connected component C of $G[V \setminus S]$ if and only if there exists a minimal triangulation $M = (V, E \cup F'')$ sandwiched between G and H. The potential maximal clique Ω in a good triple (C, S, Ω) defines how the connected component C should be partitioned into smaller blocks. We can obtain the following formulas defining how $\alpha(C, S)$ can be computed using only values of the β function, and how $\beta(C, S, \Omega)$ can be computed using only values of the α function. First formula is quite simple, since $\alpha(C, S) = 1$ if and only if there exists a minimal triangulation of $G[C \cup S]$ where $G[S]$ is completed into a clique, which should imply that there exists a potential maximal clique Ω. Thus,

$$\alpha(C,S) = \begin{cases} 1, & \text{if } \beta(C,S,\Omega) = 1 \text{ for some good triple } (C,S,\Omega) \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

The second formula comes from the observation that Ω defines how the connected component C should be decomposed further. So there exists a sandwiched minimal triangulation of $G[C \cup S]$ where Ω is a clique if and only if there exists a sandwiched minimal triangulation of $G[C' \cup N(C')]$ for each connected component C' of $G[C \setminus \Omega]$. Thus,

$$\beta(C, S, \Omega) = \begin{cases} 1, & \text{if } \alpha(C', N(C')) = 1 \text{ for every connected component} \\ & C' \in G[C \setminus \Omega] \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Theorem 7. *Given a graph $G = (V, E)$, a graph $H = (V, E \cup F)$, and a vertex cover R of (V, F) where $k = |R|$, there is an algorithm for deciding whether there exists a chordal graph sandwiched between G and H, with running time $\mathcal{O}(2^k n^5)$. Such a solution can also be returned within the same time bound.*

Proof. By Lemmas 5 and 6 there are at most $2(2^k \cdot n^2)$ possible vertex sets that are either a minimal separator or a potential maximal clique in G and a clique in H, and the two lists can be obtained in $\mathcal{O}(2^k \cdot n^5)$ time (Lemmas 5 and 6).

Let $\mathcal{P}$ be the set of potential maximal cliques in G that induces cliques in H. Let $\mathcal{B}, \mathcal{T}, \mathcal{S}$ be empty sets storing respectively blocks, good triples, and minimal separators. Remember that $\mathcal{P}$ contains all the potential maximal cliques used in a potential solution, but it remains to check if a complete clique tree of a minimal triangulation can actually can be build. For each potential maximal clique $\Omega \in \mathcal{P}$, and for each connected component C of $G[V \setminus \Omega]$, let C' be the unique connected component of $G[V \setminus N[C]]$ such that $C' \cap \Omega \neq \emptyset$. Add the minimal separator $N(C')$ to $\mathcal{S}$, add the full block $(C', N(C'))$ to $\mathcal{B}$, and add the good triple $(C', N(C'), \Omega)$ to $\mathcal{T}$. Add also a bidirectional reference from the minimal separator $S = N(C') \in \mathcal{S}$ to the full block $(C', N(C'))$ in $\mathcal{B}$ and the good triple $(C', N(C'), \Omega)$ in $\mathcal{T}$. As a result a minimal separator $S \in \mathcal{S}$ will have a reference to every full block $(C, N(C))$ where C is a connected component of $G[V \setminus S]$, and a reference to every good triple $(C, N(C), \Omega)$. Sort the full blocks and the good triples into n bins on increasing cardinality of C.

Now we will use dynamic programming to compute $\alpha(C, S)$ for all full blocks (C, S) of G and $\beta(C, S, \Omega)$ for all good triples (C, S, Ω) in the claimed time bound.

Recall that checking if there exists a chordal graph sandwiched between G and H is equivalent to checking if there exists a minimal triangulation sandwiched between G and H. A minimal triangulation is a chordal graph, and every chordal graph has a clique tree representation, and we will now check if a clique tree for such a minimal triangulation can be built from the set of potential maximal cliques in $\mathcal{P}$. A leaf of a clique tree has the property that it can be defined by a single vertex only contained in the leaf clique of the tree. Let x be a vertex only contained in a leaf clique, then $G[V \setminus N[x]]$ contains a single connected component C, and $S = N(C)$ is a minimal separator. Let C' be the connected component of $G[V \setminus S]$ such that $x \in C'$. Vertex x defines block (C', S) and the good triple (C', S, Ω) where $\Omega = S \cup C'$. If $(C', S) \in \mathcal{B}$ set $\alpha(C', S) = 1$, and 0 otherwise. If $(C', S, \Omega) \in \mathcal{T}$ set $\beta(C', S, \Omega) = 1$, and 0 otherwise. Completing this for every $x \in V$ ensures that the value for α and β is set for all blocks and good triples representing leafs. Now on increasing size of the connected component C, compute $\beta(C, S, \Omega)$ using Equation 2 which only require values of $\alpha(C', S')$ where $C' \subset C$ and $|C'| < |C|$. If the value of $\beta(C, S, \Omega) = 1$, then set $\alpha(C, S) = 1$, like stated in Equation 1. Finally if $\alpha(C, N(C)) = 0$ for at least one connected component C of $G[V \setminus S]$ for every minimal separator S of G, then "no" can be returned since no chordal graph is sandwiched between G and H. In the opposite case backtracking, using the α and β functions enables us to return a triangulation $M = (V, E \cup F')$.

For the time analysis, building the lists $\mathcal{S}, \mathcal{B}, \mathcal{T}$ is linear in $\mathcal{P}$ and requires $\mathcal{O}(nm)$ time to find the connected components, and add the references. Computing the values for the leafs requires $\mathcal{O}(nm)$ time since we find connected components starting from each vertex of the graph. Computing α and β functions for larger components can be done in $\mathcal{O}(n^2)$ time due to the references. Finally we test for each minimal separator, which there are at most $2^k \cdot n^2$ of and for each of these we can check in $\mathcal{O}(n^2)$ time by the references whether their α-function is set to 1. In total this will require $\mathcal{O}(2^k n^5)$ time is required, since $|\mathcal{P}| \leq 2^k \cdot n^2$. □

5 Concluding remarks

A small vertex cover of the admissible edges does not give any restriction to the input graph G, and it restricts the set F of admissible edges in a less strict way than for example $|F|$. Surprisingly it suffices as a parameter to achieve fixed parameter tractability of the Chordal Sandwich problem in a non-trivial way. We find this interesting, especially considering that the problem (even a more restricted version of it) remains W-hard when parameterized by a powerful parameter like the treewidth of the input graph.

Since we proved that k-Chordal Sandwich is FPT, a natural open problem is whether it admits a small kernel. Kernels that are of size polynomial in the parameter have attracted increasing attention recently, so it would be natural to look for such a kernel. However it might also be possible to prove that the problem does not admit a polynomial kernel thanks to the new tools developed in [4]. In this case also small exponential kernels or "cheat" kernels [11] become interesting. We suspect that k-Chordal Sandwich might not admit a polynomial kernel.

Another interesting question is what other parameter concerning the set of admissible edges could be natural for Chordal Sandwich, with respect to a non-trivial and efficient parameterized algorithm.

Finally we would like to mention that the algorithm could easily be adapted to finding the chordal graph sandwiched between G and H, using the smallest amount of edges from F. In the case where F is exactly the set of non-edges of G, this is equivalent to the MINIMUM FILL-IN problem.

References

[1] H. L. Bodlaender, M. R. Fellows, and T. Warnow. Two strikes against perfect phylogeny. *ICALP 1992, LNCS*, 623:273283, 1992.

[2] H. .L. Bodlaender, M. R. Fellows, M. T. Hallett, T. Wareham, and T. Warnow. The hardness of perfect phylogeny, feasible register assignment and other problems on thin colored graphs. *Theor. Comput. Sci.* 244(1-2):167–188, 2000.-155.

[3] H. L. Bodlaender, P. Heggernes, and Y. Villanger. Faster parameterized algorithms for Minimum Fill-In. *ISAAC 2008, LNCS*, 5369:282-293.

[4] H. L. Bodlaender, R. G. Downey, M. R. Fellows, D. Hermelin. On Problems without Polynomial Kernels (Extended Abstract). *ICALP 2008, LNCS*, 5125:563–574.

[5] V. Bouchitté and I. Todinca. Treewidth and minimum fill-in: Grouping the minimal separators. *SIAM Journal on Computing* 31:212–232, 2002.

[6] P. Buneman. A characterization of rigid circuit graphs. *Disc. Math.*, 9:205–212, 1974.

[7] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.*, 58:171-176, 1996

[8] M. R. Cerioli, H. Everett, C. M. Herrera de Figueiredo, and S. Klein. The homogeneous set sandwich problem. *Inf. Process. Lett.*, 67(1):3135, 1998.

[9] G. A. Dirac, On rigid circuit graphs, *Anh. Math. Sem. Univ. Hamburg*, 25:71–76, 1961.

[10] R. G. Downey and M. R. Fellows. *Parameterized complexity*, Monographs in Computer Science, Springer-Verlag, 1999.

[11] H. Fernau, F. Fomin, D. Lokshtanov, D. Raible, S. Saurabh and Y. Villanger. Kernel(s) for Problems With no Kernel: On Out-Trees With Many Leaves. *STACS 2009, Dagstuhl Seminar Proceedings*, 09001:421–432.

[12] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Springer-Verlag, 2006.

[13] F. V. Fomin, D. Kratsch, I. Todinca, and Y. Villanger. Exact algorithms for treewidth and minimum fill-in. *SIAM J. Computing*, 38:1058-1079, 2008.

[14] F. V. Fomin and Y. Villanger. Treewidth computation and extremal combinatorics. *ICALP 2008, LNCS*, 5125:210-221.

[15] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Second edition. Annals of Discrete Mathematics 57. Elsevier, 2004.

[16] M. C. Golumbic, H. Kaplan, and R. Shamir. Graph sandwich problems. *J. Algorithms*, 19(3):449–473, 1995.

[17] M. C. Golumbic and A. Wassermann. Complexity and algorithms for graph and hypergraph sandwich problems. *Graphs and Combinatorics*, 14:223–239, 1998.

[18] M. Habib, D. Kelly, E. Lebhar, and C. Paul. Can transitive orientation make sandwich problems easier? *Disc. Math.*, 307(16):2030–2041, 2007.

[19] P. Heggernes. Minimal triangulations of graphs: A survey. *Disc. Math.*, 306:297–317, 2006.

[20] P. Heggernes, J. A. Telle, Y. Villanger. Computing Minimal Triangulations in Time $O(n^{\alpha} \log n) = o(n^{2.376})$. *SODA 2005*: 907–916.

[21] C. M. Herrera de Figueiredo, L. Faria, S. Klein, and R. Sritharan. On the complexity of the sandwich problems for strongly chordal graphs and chordal bipartite graphs. *Theor. Comput. Sci.*, 381(1-3):57–67, 2007.

[22] C. M. Herrera de Figueiredo, S. Klein, and K. Vuskovic. The graph sandwich problem for 1-join composition is NP-complete. *Disc. Appl. Math.*, 121(1-3):73–82, 2002.

[23] H. Kaplan, R. Shamir, and R. E. Tarjan. Tractability of parameterized completion problems on chordal and interval graphs: Minimum fill-in and physical mapping. *FOCS 1994*:780–791.

[24] H. Kaplan, R. Shamir, and R. E. Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM J. Computing*, 28:1906–1922, 1999.

[25] D. Lokshtanov. On the Complexity of Computing Treelength. *MFCS 2007*.

[26] D. Marx. Chordal Deletion Is Fixed-Parameter Tractable. *Algorithmica*, To appear.

[27] A. Natanzon, R. Shamir, and R. Sharan. A polynomial approximation algorithm for the minimum fill-in problem. *SIAM J. Computing*, 30:1067–1079, 2000.

[28] R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, 2006.

[29] A. Parra and P. Scheffler, Characterizations and algorithmic applications of chordal graph embeddings, *Disc. Appl. Math.*, 79:171–188, 1997.

[30] D. J. Rose, R. E. Tarjan, and G. S. Lueker, Algorithmic aspects of vertex elimination on graphs, *SIAM J. Comput.*, 5:266–283, 1976.

[31] M. Schaefer. The graph sandwich problem for a coNP property. Tech. Report TR 06-011, DePaul University, 2006.

[32] M. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *J. of Classification*, 9:91–116, 1992.