REPORTS
IN
INFORMATICS

ISSN 0333-3590

Minimum fill-in and treewidth of split+ke and split+kv graphs

Federico Mancini

REPORT NO 355 **June 2007**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL http://www.ii.uib.no/publikasjoner/texrap/ps/2007-355.ps

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at
http://www.ii.uib.no/publikasjoner/texrap/.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

Minimum fill-in and treewidth of split+ke and split+kv graphs*

Federico Mancini †

Abstract

In this paper we investigate how graph problems that are NP-hard in general, but polynomially solvable on split graphs, behave on input graphs that are close to being split. For this purpose we define split+ke and split+kv graphs to be the graphs that can be made split by removing at most k edges and at most k vertices, respectively. We show that problems like treewidth and minimum fill-in are fixed parameter tractable with parameter k on split+ke graphs. Along with positive results of fixed parameter tractability of several problems on split+ke and split+kv graphs, we also show a surprising hardness result. We prove that computing the minimum fill-in of split+kv graphs is NP-hard even for $k = 1$. This implies that also minimum fill-in of chordal+kv graphs is NP-hard for every k. In contrast, we show that the treewidth of split+$1v$ graphs can be computed in polynomial time. This gives probably the first graph class for which the treewidth and the minimum fill-in problems have different computational complexity.

1 Introduction

Many NP-hard graph problems become polynomially solvable when restricted to specific graph classes. Let $\mathcal{C}$ be such a class. A natural question is whether a problem that is tractable on $\mathcal{C}$ remains tractable when we consider a graph class that is close to $\mathcal{C}$. For example a class where every graph can be made into a graph of $\mathcal{C}$ altering only few edges or vertices. This is a common situation in practical applications, where the input can be affected by errors or incomplete information. For this purpose we define the graph classes $\mathcal{C}+ke$, $\mathcal{C}-ke$ or $\mathcal{C}+kv$ to be the class of graphs that can be obtained by, respectively, adding at most k edges, removing at most k edges or adding at most k vertices to the graphs in $\mathcal{C}$. Notice that when dealing with hereditary properties, it does not make sense to remove vertices. We will often refer to these classes of graphs as *parametrized graph classes*.

Recently several authors studied the complexity of hard problems on such graph classes from a parametrized point of view [4, 5, 29, 22, 11]. In particular in [11] the whole idea is generalized and the parameter k is considered as a measurement of the "distance from triviality". In other words k is used to measure how the complexity of a problem changes as we get further from a trivial solution. For a given problem and a graph class on which it has a polynomial time solution, the main questions we can ask about the corresponding parametrized classes are: Does the problem remain polynomial time solvable up to some k and become NP-hard for $k+1$ or does it remain polynomial for each fixed k? And in the last case, how does it behave from a parametrized complexity point of view? Is it FPT or W-hard? A problem with parameter k is fixed parameter tractable (FPT) or *uniformly polynomial*, if it can be solved in time $O(f(k) \cdot |x|^c)$, where f is an arbitrary function and $|x|$ is the size of the input. When a problem is W-hard it might still have a polynomial time algorithm for each fixed k, for example $O(|x|^k)$, but it is very unlikely to be FPT. For a complete reference see [8].

A fundamental question, and an interesting problem on its own, is whether it is possible to recognize parametrized graph classes in FPT time. Cai [4] showed that for all classes of graphs characterized by a finite forbidden set of induced subgraphs, the corresponding parametrized classes ($+kv$, $+ke$, $-ke$) can be recognized in FPT time. In addition it has been shown that also chordal+kv [22], chordal$-ke$ [20, 21, 4], strongly chordal$-ke$ [20], interval$-ke$ [17], proper interval$-ke$ [20, 21] and planar+kv graphs [24] are recognizable in FPT time. Split graphs are characterized by a finite forbidden set of

*This work is supported by the Research Council of Norway through grant 166429/V30.
†Department of Informatics, University of Bergen, N-5020 Bergen, Norway. Email: federico@ii.uib.no

induced subgraphs, hence, by the result of [4], split+ke, split−ke and split+kv graphs are recognizable in FPT time. By the same result we can assume that, given our input, we know which set of k edges or vertices we have to remove/add to get a split graph. Such a set is called a *modulator* of the graph. Related to the problem of finding a modulator efficiently, in [5] Cai asks whether there is a recognition algorithm that is not only uniformly polynomial, but uniformly linear for parametrized split graphs. We can answer affirmatively to this question.

At the moment it seems like mostly coloring problems have been investigated on parametrized graph classes. In [5], Cai shows that finding the chromatic number of split+ke and split−ke graphs is FPT, while it is W[1]-hard for split+kv graphs. He also shows that the same problem is solvable in linear time on bipartite+1v and bipartite+2e graphs, but NP-complete for bipartite+2v and bipartite+3e graphs. Takenaga and Higashide [29] study the chromatic number problem for comparability+ke graphs, and prove that it becomes NP-complete for $k \geq 2$. Finally in [23], Marx gives an FPT algorithm for coloring chordal+ke graphs.

In this paper we consider various other problems that are known to be polynomially solvable on split graphs, and we study their complexity for split+ke and split+kv graphs. For example we study problems like minimum split completions and minimum fill-in. In these cases we want to find the minimum number of edges to be added to a graph to make it split or chordal. Minimum fill-in in particular is an extremely well studied problem with a number of practical applications (see for example [13]). Of course these problems are easy on split graphs since split graphs are chordal. That is why it comes as a great surprise that, even though minimum split completion of split+kv graphs is solvable in FPT time, minimum fill-in becomes NP-complete even for split+1v graphs. This implies that minimum fill-in of chordal+1v graphs is also NP-complete. Motivated by this result we investigate also the treewidth of split+kv graphs. Threewidth is another problem related to chordal completions with important algorithmic applications (see [3] for a survey). In contrast with the previous result, we prove that treewidth of split+1v graphs can be computed in polynomial time, giving the first graph class (to our knowledge) on which minimum fill-in and treewidth have different computational complexity.

It is worth mentioning that there has been some work on similar problems, but restricted to special cases: minimum split completions of split+1v and +1e graphs [16] and minimum cograph completions of cograph+1e graphs [25].

In section 3 we give a simple FPT algorithm to list all maximal independent sets and cliques of a split+kv graphs. This leads to the FPT algorithm for computing the minimum split completion of split+kv graphs, generalizing the result of [16]. Let us also point out that all FPT algorithms given for split+kv graphs, work also for split+ke graphs, but not vice versa.

In section 4 we focus on minimum fill-in and treewidth for split+ke and split+kv graphs. As far as split+ke graphs are concerned, we are able to give an FPT algorithm that given a graph in this class, can list all its minimal triangulations. This implies that we can compute both treewidth and minimum fill-in within the same time bound. For split+kv graphs we prove the results mentioned above.

We conclude with listing some open problems and possible directions of research.

2 Notation and definitions

All graphs in this paper are simple and undirected. For a graph $G = (V, E)$, we let $n = |V|$ and $m = |E|$. The set of neighbors of a vertex $v \in V$ is denoted by $N(v)$, and the degree of a vertex v is denoted by $d(v) = |N(v)|$. The neighborhood of a set of vertices S is defined as $N(S) = \cup_{x \in S} N(x) \setminus S$. Also, $N[v] = N(v) \cup \{v\}$ and $N[S] = \cup_{x \in S} N[x]$. We distinguish between subgraphs and induced subgraphs. In this paper, a *subgraph* of $G = (V, E)$ is a graph $G_1 = (V, E_1)$ with $E_1 \subseteq E$, and a *supergraph* of G is a graph $G_2 = (V, E_2)$ with $E \subseteq E_2$. We will denote these relations informally by the notation $G_1 \subseteq G \subseteq G_2$ (proper subgraph relation is denoted by $G_1 \subset G$). An induced subgraph of $G = (V, E)$ over a set of vertices $U \subseteq V$, is the graph $G[U] = (U, E_U)$, where $E_U = \{xv \in E | x, v \in U\}$. The complement of G is denoted by $\overline{G}$. Given a vertex x of G, $G - x = G[V \setminus \{x\}]$.

A subset K of V is a *clique* if K induces a complete subgraph of G. A subset I of V is an *independent set* if no two vertices of I are adjacent in G. We use $\omega(G)$ to denote the size of the largest clique in G, and $\alpha(G)$ to denote the size of a largest independent set in G. If there is no ambiguity, we will use only ω and α. We call a vertex v *simplicial* if $N(v)$ induces a clique. A vertex cover is a set $V' \subseteq V$ such that all edges of G are incident to at least a vertex in V'. A dominating set, is a set $V' \subseteq V$ such that $N[V'] = V$.

G is a *split graph* if there is a partition $V = I + K$ of its vertex set into an independent set I and a clique K. Such a partition is called a *split partition* of G. There is no restriction on the edges between vertices of I and vertices of K. The partition of a split graph into a clique and an independent set is not necessarily unique. The following theorem states the possible partition configurations.

Theorem 2.1. (Hammer and Simeone [12]) *Let G be a split graph whose vertices have been partitioned into an independent set I and a clique K. Exactly one of the following conditions holds:*

(i) $|I| = \alpha(G)$ *and* $|K| = \omega(G)$
(in this case the partition $I + K$ is unique),

(ii) $|I| = \alpha(G)$ *and* $|K| = \omega(G)-1$
(in this case there exists a vertex $x \in I$ such that $K \cup \{x\}$ is a clique),

(iii) $|I| = \alpha(G)-1$ *and* $|K| = \omega(G)$
(in this case there exists a vertex $y \in K$ such that $I \cup \{y\}$ is independent).

For the following result, note that a simple cycle on k vertices is denoted by C_k and that a complete graph on k vertices is denoted by K_k. Thus $2K_2$ is the graph that consists of 2 isolated edges. Also, a chordal graph is a graph that does not contain an induced subgraph isomorphic to C_k for $k \geq 4$.

Theorem 2.2. (Földes and Hammer [9]) *Let G be an undirected graph. The following conditions are equivalent:*

(i) *G is a split graph.*

(ii) *G and $\overline{G}$ are chordal graphs.*

(iii) *G contains no induced subgraph isomorphic to $2K_2, C_4$ or C_5.*

Remark 2.3. *Every induced subgraph of a split graph is also a split graph.*

A graph $G = (V, E)$ is called: A split+ke graph if there is a set E_k with $E_k \subset E$ and $|E_k| \leq k$ such that $G' = (V, E \setminus E_k)$ is a split graph; A split$-ke$ graph if there is a set E_k with $E_k \cap E = \emptyset$ and $|E_k| \leq k$ such that $G' = (V, E \cup E_k)$ is a split graph; and split+kv graph if there is a set $V_k \subset V$ with $|V_k| \leq k$ such that $G[V \setminus V_k]$ is a split graph. The set E_k or V_k is referred to as a *modulator* of the graph.

For a given arbitrary graph $G = (V, E)$, a split graph $H = (V, E \cup F)$, with $E \cap F = \emptyset$, is called a *split completion* of G. The edges in F are called *fill edges*. H is a *minimum* split completion of G if $|F|$ is as small as possible, while H is a *minimal* split completion of G if $(V, E \cup F')$ fails to be a split graph for every proper subset F' of F.

Minimal and minimum chordal completions are defined analogously to minimal and minimum split completions and they are also called *triangulations*. In particular the problem of making a graph chordal adding the minimum number possible of fill edges is referred to as the minimum fill-in problem. Both minimum fill-in For all vertices $v \in V$, $\{i \in I \mid v \in X_i\}$ induces a connected subtree of T.and the minimum split completion problem are NP-hard [30, 1]. For a split or chordal graph G, $\alpha(G)$ and $\omega(G)$ can be computed in linear time [10], whereas these are NP-hard problems for general graphs.

Treewidth is a parameter that measures how tree-like a graph is, and computing it is NP-hard for general graphs [28]. The formal definition involves the concept of tree decomposition.

Definition 2.4. *A tree decomposition of a graph $G = (V, E)$ is a pair $(\{X_i \mid i \in I\}, T = (I, M))$ where $\{X_i \mid i \in I\}$ is a collection of subsets of V (also called bags), and T is a tree such that:*

(i) $\bigcup_{i \in I} X_i = V$

(ii) $(u, v) \in E \Longrightarrow \exists i \in I$ *with* $u, v \in X_i$

(iii) *For all vertices $v \in V$, $\{i \in I \mid v \in X_i\}$ induces a connected subtree of T.*

The *width* of a decomposition $(\{X_i \mid i \in I\}, T = (I, M))$ is $\max_{i \in I} |X_i| - 1$. The *treewidth* of a graph G, $tw(G)$, is the minimum width over all tree decompositions of G. The treewidth of a graph can also be defined from the point of view of minimal triangulations. In particular, the treewidth of a graph G is $\min_H \omega(H) - 1$, where H is a minimal triangulation of G. So finding the treewidth of G is equivalent to finding a minimal triangulation H of G with the smallest maximum clique. Chordal graphs have a tree decomposition with minimum width called *clique tree*, where each bag corresponds to a maximal clique and each leaf-bag contains at least a simplicial vertex [2]. Also, for a chordal graph G, we have $tw(G) = \omega(G) - 1$. Finally, as mentioned in the introduction, if a problem is FPT for split+kv graphs, then it is FPT also for split+ke graphs. In fact every split+ke graph is also a split+kv graph. To see this: we can always select at most k vertices that cover all the k edges. Removing these vertices clearly gives a split graph by Remark 2.3. Furthermore, the complement of a split+ke graph, is a split$-ke$ graph since split graphs are self complementary, which follows from Theorem 2.2. Hence, if we can solve a problem for split+ke graph, we can solve its complement on split$-ke$ graph. For example if we can solve the minimum split completion problem for split+ke graphs, then we can solve the minimum split deletion problem for split$-ke$ graphs.

3 Listing maximal independent sets in FPT time

It is already known that for every hereditary family $\mathcal{C}$ for which maximum clique, maximum independent set and minimum vertex cover are polynomial time solvable, then the same problem can be solved in FPT time on the corresponding parametrized classes [5]. However in this case we are interested in listing all maximal independent sets and cliques in FPT time, not only finding the maximum. Notice that if it is possible to list all maximal independent sets or cliques in FPT time, then it is possible possible to find the corresponding maximum as well, but not vice versa. In particular it is easy to show that for every hereditary family $\mathcal{C}$ with a polynomial number of maximal independent sets (or cliques), it is possible to list in FPT time all the maximal independent sets (or cliques) of $\mathcal{C} + kv$. In this section we will prove formally such result, and apply it to split+kv graphs. This will lead to a simple FPT algorithm to solve the minimum split completion problem for this graph class.

Before to start, we would like to settle also a question left open by Cai in [5] about the existence of a uniformly linear algorithm to find a modulator for parametrized split graphs. The answer is affirmative, and follows from [16, 14] where two linear time certification algorithms for split graphs are given. These algorithms, in fact, can not only recognize if a graph is split, but also return a forbidden subgraph if it is not. Once we can find a forbidden subgraph in linear time, we can try and remove each of its vertices, and run the recognition algorithm again for at most k times in each case. Since the largest forbidden subgraph of a split graph has 5 vertices, this procedure would yield a search tree with at most $O(5^k)$ nodes and therefore a total running time of order $O(5^k \cdot (n + m))$. The same idea holds for any graph class $\mathcal{C}$ with a finite set of forbidden induced subgraphs. A modulator of the corresponding parametrized family can be found in linear time for each fixed k if, given a graph not in $\mathcal{C}$, a forbidden subgraph can be found in linear time.

The following is implicit from the results of [19].

Lemma 3.1. *[19] If a graph has a polynomial number of maximal independent sets, then they can be listed in polynomial time.*

Lemma 3.2. *Let $\mathcal{C}$ be a hereditary family of graphs with a polynomial number of maximal independent sets. Then, given a graph $G = (V, E)$ in $\mathcal{C} + kv$ and a modulator V_k, it is possible to list all its maximal independent sets in FPT time.*

Proof. Let $V = V_s \cup V_k$. Every maximal independent set I of G can be expressed in the form $I = I_{V_k} \cup I_{V_s}$, where $I_{V_k} = I \cap V_k$, and $I_{V_s} = I \cap V_s$. Naturally, for I to be maximal, the following condition must hold: I_{V_s} is a maximal independent set of $G[V_s \setminus N(I_{V_k})]$ such that $N[I_{V_s}] \cup N[I_{V_k}] = V$. Let us call such I_{V_s} a "legal" independent set. The algorithm will therefore go as follows. For every independent set I_{V_k} in $G[V_k]$ (including the empty set), list all the maximal independent sets of $G[V_s \setminus N(I_{V_k})]$, and for each of them check whether it is legal or not. If so, return $I = I_{V_s} \cup I_{V_k}$. This will clearly produce all maximal independent sets of G. Listing all independent sets of $G[V_k]$, takes time $O(2^k \cdot k^2)$. Given one of them, I_{V_k}, it takes polynomial time to list all the maximal independent sets of $G[V_s \setminus N(I_{V_k})]$ and find the legal ones. Notice in fact that $G[V_s \setminus N(I_{V_k})] \in \mathcal{C}$, therefore it has a polynomial number of maximal independent sets and they can be listed maximal in polynomial time by Lemma 3.1. Checking whether any of them is legal, clearly takes polynomial time as well. Hence the overall running time is $O(2^k \cdot k^2 \cdot p(n))$ for some polynomial function $p(n)$. □

At this point it is enough to show that split graphs have a polynomial number of maximal independent sets to get the result we need.

Lemma 3.3. *In a split graph there are at most $\omega + 1$ maximal independent sets and $\alpha + 1$ maximal cliques.*

Proof. Let K be a maximum clique of a split graph $G = (V, E)$ with maximum clique size ω, and I the independent set induced by $V \setminus K$. For each vertex $x \in K$ there is exactly one maximal independent set of G, namely $V \setminus N(x)$. Therefore there are at least $|K| = \omega$ maximal independent sets. Besides, if $(N(x) \cap I) \neq \emptyset$ for every $x \in K$, then I is also a maximal independent set, because it cannot be extended with any vertex of K. Hence in a split graph there are at most $|K| + 1$ maximal independent sets. Having the number of maximal independent sets, the number of maximal cliques follows from the fact that split graphs are self complementary. In fact every maximal clique in G is a maximal independent set in $\overline{G}$, and being split graph perfect we have that $\alpha(G) = \omega(\overline{G})$. □

Theorem 3.4. *Let $G = (V, E)$ be a split+kv graph, and let V_k be a modulator of G. Then the maximal independent sets of G can be listed in time $O(2^k \cdot k^2 \cdot nm)$.*

Proof. To prove the given running time, we show a possible implementation of the proof of Lemma 3.2 for split graphs. Given an independent set I_{V_k} of $G[V_k]$, let $V_s = V \setminus V_k$ and let us define $V'_s = V_s \setminus N(I_{V_k})$. We store the graph as an adjacency list where for each neighbor u in the list of a vertex v, we keep a double pointer to the position of v in the list of u. Also we store all the vertices of V_k before V_s. In this way, finding $V_k \setminus I_{V_k}$ and removing all neighbors of a given set of vertices can be done in linear time. Hence the graph $G[V'_s]$ can be found in linear time. All the maximal independent sets of the split graph $G[V'_s]$ can be listed in $O(n^2)$ time as follows: Find a maximum clique K in linear time, and for each vertex $v \in K$ find $V \setminus N(v)$ in $O(n)$ time; then check if $V'_s \setminus K$ is a dominating set for $G[V'_s]$ in linear time. Finally, for every maximal independent set I_{V_s}, we can check whether $I_{V_k} \cup I_{V_s}$ is a dominating set for G in linear time. In total we have $O(n + m + n^2 + nm) = O(nm)$. □

Since all previous results on maximal independent sets hold also for maximal cliques (just consider the complement of the graph instead), we can give the following corollary.

Theorem 3.5. *Let $G = (V, E)$ be a split+kv graph. Then a maximum independent set, maximum clique, minimum vertex cover or minimum independent dominating set of G can be found in time $O(2^k \cdot k^2 \cdot nm)$.*

In the rest of the section we give an FPT algorithm to compute a minimum split completion of split+kv graphs. In order to prove the result, we use the characterization of minimal split completions given in [15], that is based on the 3-partition of split graphs introduced in the same paper.

The 3-partition of the vertices V of a split graph $G = (V, E)$ consists in three sets $V = S + C + Q$ defined as following:

$$S = \{v \in V \mid d(v) < \omega(G) - 1\} \qquad C = \{v \in V \mid d(v) > \omega(G) - 1\} \qquad Q = \{v \in V \mid d(v) = \omega(G) - 1\}$$

This partition is obviously unique, however it has been defined only to overcome the ambiguity caused by the many possible split partitions a split graph can have; therefore, when a graph has a unique split partition $V = I + K$, we set $S = I$, $C = K$ and $Q = \emptyset$. Given the previous partition, we can then characterize minimal split completions.

Lemma 3.6. *[15] Let $H = (V, E + F)$ be a split completion of an arbitrary graph $G = (V, E)$, and let $V = S + C + Q$ be the 3-partition of H. Then H is a minimal split completion of G if and only if each fill edge has both its endpoints in C.*

Lemma 3.7. *[15] Let $G = (V, E)$ be a split graph, and $V = S + C + Q$ the 3-partition of G. Then $S \subseteq I$ and $C \subseteq K$, for every split partition $V = I + K$ of G.*

We are now ready to give the main lemma.

Lemma 3.8. *Let $H = (V, E \cup F)$ be a minimal split completion of a graph $G = (V, E)$. Then there exists a maximal independent set $I \subseteq V$ in G such that H can be obtained making $G[V \setminus I]$ into a clique.*

Proof. Let $V = S + C + Q$ be the 3-partition of H. Since H is minimal, we know by Theorem 3.6 that all fill edges are in C. Take now a split partition $V = K + I$ of H such that I is a maximum independent set of H. This is always possible by Theorem 2.1. Since $C \subseteq K$ for every K by Lemma 3.7, there are no fill edges between I and K, and I dominates K since it is maximum. This means that I dominates K in G as well, therefore it is a maximal independent set of G. Since adding F to $G[V \setminus I]$ we get exactly the clique K, the lemma follows. □

Notice that it is not true, however, that a minimum split completion of a graph $G = (V, E)$ can be obtained taking some maximum independent set I of G and making $G[V \setminus I]$ into a clique. To see this. Take a complete bipartite graph $G_{AB} = (A \cup B, E)$ and a clique C. Build a graph G adding all edges between the vertices of A and C. Let $|A| = 2|B|$, and $|C| = c$. The maximum independent set of G is A, but making $G[C \cup B]$ into a clique requires $O(c \cdot |B|^2)$ edges, while making $G[C \cup A]$ into a clique requires $O(|A|^2)$ edges. Just choose c big enough $(O(|A|))$ to have the counter example. Actually we cannot even guarantee that taking a maximum independent set we can produce a minimal split completion (consider a path on 4 vertices). That is why we need to list all maximal independent sets of a graph to find a minimum split completion of it.

Theorem 3.9. *Let $G = (V, E)$ be a split+kv graph. Then a minimum split completion of G can be computed in time $O(2^k \cdot k^2 \cdot nm)$.*

Proof. By Theorem 3.4 we know that we can list all maximal independent sets of G in time $O(2^k \cdot k^2 \cdot nm)$, and by Lemma 3.8 we know that having all maximal independent sets, we can produce all minimal split completions of G. Hence finding the minimum requires only to choose the maximal independent set I that minimizes the number of non edges in $G[V \setminus I]$, among the ones listed. □

Using Lemma 3.8, it is also straightforward to give an exact algorithm for minimum split completions.

Corollary 3.10. *A minimum split completion of a graph $G = (V, E)$ can be found in time $O^*(3^{|V|/3})$.*

Proof. It follows from Lemma 3.8 and the fact that all maximal independent set of a graph can be listed in $O^*(3^{|V|/3})$ time by [19]. □

4 Minimum fill-in and treewidth

In this section we give an FPT algorithm for minimum fill-in and treewith of split+ke graphs, and show that the same problem is harder for split+kv graphs. In particular we prove that there exists a graph class for which minimum fill-in is NP-complete while treewidth is polynomial, namely split+$1v$ graphs.

4.1 Split+ke graphs

In this section we will use the connection between the Elimination Game and chordal graphs [27]. Running the Elimination Game on a graph $G = (V, E)$ and an ordering β of its vertices, means to remove the vertices from G in the order given by β so that, after removing a vertex, we make its neighborhood in the current graph into a clique. It is well known that this produces a triangulation of G. In particular for each minimal triangulation H of a graph G, there exists an ordering β that can produce it [26]. An ordering is called *perfect elimination ordering* if every vertex is simplicial when it is deleted during the Elimination Game. Chordal graphs are exactly the graphs that admit a perfect elimination ordering.

We will show that for split+ke graphs, we can produce in FPT time all elimination orderings that produce a minimal triangulation.

The results in Observation 4.1 and 4.2 follow from previous results on chordal graphs and minimal triangulations, but, because of their simplicity, we include a small proof anyway (for further references see [13]).

Observation 4.1. *Given a graph $G = (V, E)$, let $K \subseteq V$ be a set of vertices such that $G[K]$ is a clique. Then every minimal triangulation H of G can be obtained by running the Elimination Game on G and an elimination ordering where the vertices of K are eliminated at the end, in any order.*

Proof. It is enough to show that there exists a perfect elimination ordering of such form for every minimal triangulation of G. Let H be a minimal triangulation of G and let us assume that H is not a clique, or the result is trivial. Clearly if $G[K]$ is a clique, $H[K]$ is also a clique. Hence, since H is chordal and $K \subset V$, there exists a simplicial vertex in $H[V \setminus K]$, because every chordal graph is either a clique and all vertices are simplicial, or it contains at least two non adjacent simplicial vertices [7]. Remove this simplicial vertex and apply the same argument until only $H[K]$ is left. At this point all the remaining vertices are simplicial so all the elimination orderings are equivalent and the result follows. □

Observation 4.2. *Let H be a minimal chordal completion of an arbitrary graph G. No simplicial vertex of G is incident to any fill edge in H.*

Proof. Let v be a simplicial vertex in G, with $N_G(v) = T$. Assume v is incident to some fill edges in H. Clearly $H - v$ is still a chordal graph. Let us now create a new graph H' adding back v to $H - v$, but making it incident only to T. H' is still chordal because T is a clique and we cannot create any cycle making v incident only to it. Besides $G \subseteq H' \subset H$, contradicting that H is minimal. □

Using the previous two Lemmas, we will show that all minimal chordal completion of a split+ke graph, can be obtained considering all the permutations of at most $2k$ vertices.

Theorem 4.3. *Let $G = (V, E)$ be a split+ke graph, and let the set $E_k \subset E$ be a modulator for G. Then all minimal chordal completions of G can be computed in $O((2k)! \cdot nm)$ time.*

Proof. Let I be a maximum independent set of the split graph $G' = (V, E \setminus E_k)$. Notice that all vertices in I are simplicial in G'. Let K be the clique $G'[V \setminus I]$. In G, all vertices of I that are not incident to an edge of E_k are still simplicial. Hence, since $|E_k| \leq k$, there can be at most $2k$ non simplicial vertices in G that belong to I. Let us call this set S. By Observation 4.2, we can remove all vertices in $I \setminus S$ and consider only $G_S = G[S \cup K]$. $G_S[K]$ is a clique, so by Observation 4.1 all elimination orderings of G_S can be produced by considering only the orderings of the vertices in S. This means that there are at most $(2k)!$ meaningful orderings, and for each of them it takes $O(nm)$ time to both compute the corresponding triangulation and check whether it is minimal [6]. □

Corollary 4.4. *Minimum fill-in and treewidth of split+ke graphs can be computed in time $O((2k)! \cdot nm)$.*

4.2 Split+kv graphs

4.2.1 Minimum fill-in

Here we show that adding the minimum number of edges to make a graph that is split+kv into chordal, is NP-complete even for $k = 1$. In order to prove this result, we give a reduction from the minimum fill-in for co-bipartite graphs, that was shown NP-complete by Yannakakis in [30].

Let $G = (P, Q, E)$ be a co-bipartite graph, where P and Q are cliques. From G we build a new graph $G_S = (P \cup Q \cup C \cup \{x\}, E_S)$ in the following way. Take a copy of G. Remove all edges between vertices in P. Create a clique C of size $|P|(|P| - 1)/2 + |P| \cdot |Q| + 1$ and add all edges between the vertices in C and $P \cup Q$. Finally add a vertex x and make it universal to P. Since we can partition G_S into a clique $C \cup Q$, an independent set P and a vertex x, we have the following observation.

Observation 4.5. *The graph G_S is split+$1v$.*

Lemma 4.6. *In every minimum triangulation H_S of $G_S = (P \cup Q \cup C \cup \{x\}, E_S)$, $H_S[P]$ is a clique.*

Proof. First of all notice that there is a trivial upper bound for the size of a minimum triangulation of G_S, namely $|P|(|P| - 1)/2 + |P| \cdot |Q|$. That is, make $G_S[P \cup Q]$ into a clique. Furthermore, for every pair of vertices $p_1, p_2 \in P$ and every vertex $c \in C$, $G_S[\{x, p_1, c, p_2\}]$ is a C_4, and there are only two ways to kill such a cycle: either add the edge p_1p_2, or make x universal for C. We will prove the statement by contradiction. Assume there exists a minimum triangulation H'_S of G_S where $H'_S[P]$ is not a clique. Then x must be universal to C or, by the previous discussion, for any two non adjacent vertices in P and a vertex of C not incident to x, we would have a C_4 in H'_S. Notice that making x universal to C requires the addition of $|C|$ fill edges. However, by the construction of G_S, $|C| = |P|(|P| - 1)/2 + |P| \cdot |Q| + 1 > |P|(|P| - 1)/2 + |P| \cdot |Q|$. This shows that H'_S cannot be a minimum triangulation of G_S, since it exceeds the upper bound we gave previously. □

Lemma 4.7. *A minimum triangulation of a co-bipartite graph $G = (P, Q, E)$ can be obtained adding a set F of fill edges to G if and only if a minimum triangulation of $G_S = (P \cup Q \cup C \cup \{x\}, E_S)$ can be obtained adding the set of fill edges F to G_S and making $G_S[P]$ into a clique.*

Proof. From Lemma 4.6 we know that in every minimum triangulation of G_S, the subgraph induced by P is a clique. Let us then consider the graph G'_S, namely the graph G_S where $G_S[P]$ has been made into a clique. In G'_S the vertex x is simplicial, hence by Observation 4.2 there will be no fill edge incident to it. This also implies that no fill edges can be incident to any vertex of C, because they are universal for everything but x. We can conclude that finding a minimum triangulation of G_S is equivalent to finding a triangulation of $G'_S[P \cup Q]$. The fact that $G'_S[P \cup Q] = G$ concludes the proof. □

Given the previous Lemma and the fact that G_S can be built in polynomial time, we can state the main theorem of this section.

Theorem 4.8. *The minimum fill-in problem for split+kv is NP-complete for $k \geq 1$.*

Corollary 4.9. *The minimum fill-in problem for chordal+kv is NP-complete for $k \geq 1$.*

4.2.2 Treewidth

In contrast to the minimum fill-in problem, we show that the treewidth of split+kv can be found in polynomial time when $k = 1$. We have strong evidences that this still holds for $k = 2$, but for larger values of k we only conjecture the existence of a polynomial algorithm, probably not FPT.

Let $G = (V, E)$ be a split+kv graph with modulator V_k and let us define $\omega = \omega(G[V \setminus V_k])$. Since split graphs are chordal, we know that $tw(G[V \setminus V_k]) = \omega - 1$. Besides, adding k vertices to a graph cannot increase the treewidth by more than k, therefore we have that $\omega - 1 \leq tw(G) \leq \omega + k - 1$. Let us now consider $G = (V, E)$ to be a split+$1v$ graph, $V_k = \{x\}$ its modulator and again $\omega = \omega(G - x)$. Then we have the following observation.

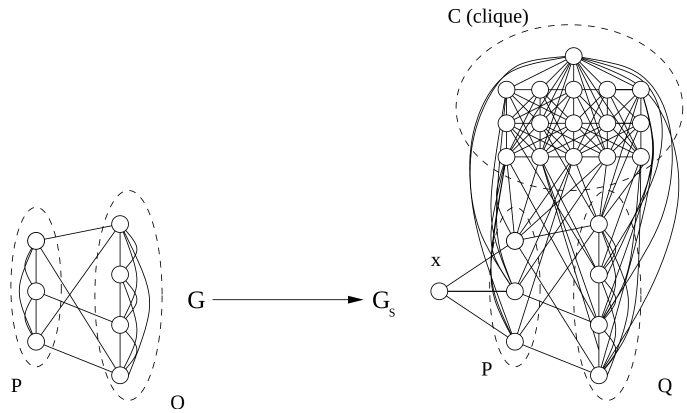

Figure 1: *Example of reduction.*

Observation 4.10. *The treewidth of split+1v graph is either* $\omega - 1$ *or* ω.

For the rest of the section we assume G not to be chordal and $\omega = \omega(G - x) = \omega(G)$. Also, we define G' to be the graph obtained from G removing recursively all simplicial vertices not in $N_G[x]$. We will prove that computing the threewidth of G is equivalent to computing the size of the maximum clique of $G' - x$.

Observation 4.11. *All simplicial vertices removed to obtain* G' *have degree at most* $\omega - 1$.

Lemma 4.12. $tw(G') < \omega$ *if and only if* $tw(G) < \omega$.

Proof. If $tw(G) < \omega$, it is straightforward to see that $tw(G') < \omega$, since G' is an induced subgraph of G. For the other direction, let H' be a minimal triangulation of G' with treewidth $< \omega$. Let us add back to H' the simplicial vertices removed to obtained G' and call the resulting graph H. Notice that H is still chordal, namely a chordal completion of G, and it does not contain cliques of size greater than ω. In fact $\omega(H') \leq \omega$ and the vertices we put back in order to get H do not have degree greater than $\omega - 1$ by Observation 4.11, so they cannot create a clique of size greater than ω. This means that there exists a chordal completion of G with maximum clique size not greater than ω. Hence $tw(G) < \omega$. □

Lemma 4.13. *Let* $\omega' = \omega(G' - x)$. *Then* $tw(G') = \omega'$.

Proof. If $\omega(G') = \omega' + 1$ the result follows directly by Observation 4.10 since G' is still a split+1v graph, otherwise take a split partition $K + I$ of $G' - x$, such that K is a clique of size ω'. Notice that x is adjacent to all I, or there would be simplicial vertices not adjacent to x. Let us assume for the sake of contradiction that the treewidth of G' is not ω'. Then it must be $\omega' - 1$ by Observation 4.10. Take a minimal triangulation H of G' with treewidth $\omega' - 1$, and a clique tree T_H of H. In H there cannot be cliques of size greater than ω'. This means that there must be a bag of T_H containing the whole clique K, and nothing else, since every clique must be completely contained in some bag of the tree decomposition. Assume such bag is an internal bag. Since H cannot be a complete graph, every internal bag of its tree decomposition is a separator in the graph [18]. However K cannot separate any two vertices in the graph, since $H[V \setminus K] = H[x \cup I]$ is a connected graph, as $G'[x \cup I]$ was. Hence the bag containing K must be a leaf of T_H. This implies that there exists a simplicial vertex $v \in H[K]$. Since G' has no simplicial vertices that are not neighbors of x, every vertex in K must have at least a neighbor in $I \cup \{x\}$. Hence $N_H[v] \supset H[K]$, meaning that there is a clique of size greater than ω' in H, giving a contradiction. □

We are now ready to give the main Theorem.

Theorem 4.14. *Treewidth is polynomial for split+1v graphs.*

Proof. Given a split+1v graph G with modulator $\{x\}$, we can check whether it is chordal in linear time [13] and, if so, output $tw(G) = \omega(G)-1$. Otherwise, if $\omega(G-x)+1 = \omega(G)$ we output $tw(G) = \omega(G-x)$ by Observation 4.10. Notice that we can find $\omega(G)$ in polynomial time by Lemma 3.5. If none of the previous holds, we can apply Lemma 4.12 and 4.13. That is, we need to find $\omega(G'-x)$ and output $tw(G) = \omega(G) - 1$ if $\omega(G'-x) < \omega(G)$, or $tw(G) = \omega(G)$ otherwise. Since G' and $\omega(G'-x)$ can be found in polynomial time, the result follows. □

4.3 Conclusions

In this paper we studied how some problems that are easy on split graphs behave when the input graph is slightly modify by the addition of some edges or vertices. The results were in part counter-intuitive and surprising, and this gives strong motivation to continue studying these parametrized graph classes. The next natural step would be to take in consideration parametrized chordal graphs, as they are a very important superclass of split graphs. For example it would be very interesting to know whether there is an FPT algorithm for minimum fill-in and treewidth for chordal+ke like for split graphs. For chordal+kv, instead, we proved that minimum fill-in is NP-complete, but we do not know much about treewidth, even though it seems polynomial for $k = 1$. However it would be maybe easier to try and settle the problem for split+kv when $k \geq 2$, since any hardness result would still hold for chordal+kv. One more problem that would be worth considering for split+kv is pathwidth, for its duality with minimum interval completions. Pathwidth is equivalent to finding a minimal interval completion with minimum clique size, and it is polynomial time solvable for split graphs, while minimum interval completion of split graphs is NP-complete. Does the same dichotomy holds when adding k vertices, or does pathwidth become hard as well?

References

[1] R. Shamir A. Natanzon and R. Sharan. Complexity classification of some edge modification problems. *Discrete Applied Mathematics*, 113(1):109–128, 2001.

[2] J. R. S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. In *Graph Theory and Sparse Matrix Computations*, pages 1–29. Springer-Verlag, 1993.

[3] H. L. Bodlaender. Discovering treewidth. In *31st Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM 2005, LNCS Proceedings*, volume 3381, pages 1–16. Springer.

[4] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.*, 58(4):171–176, 1996.

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

[6] E. Dahlhaus. Minimal elimination ordering inside a given chordal graph. In *23rd International Workshop on Graph-Theoretic Concepts in Computer Science, WG 1997, LNCS Proceedings*, volume 1335, pages 132–143. Springer.

[7] G. A. Dirac. On rigid circuit graphs. In *Abhandlungen aus dem Mathematischen Seminar der Universiat Hamburg*, volume 25, pages 71–75. University of Hamburg, 1961.

[8] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.

[9] S. Foldes and P. L. Hammer. Split graphs. In *8th South-Eastern Conference on Combinatorics, Graph Theory and Computing*, pages 311–315, 1977.

[10] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, 1st edition, 1980.

[11] J. Guo, F. Hüffner, and R. Niedermeier. A structural view on parameterizing problems: Distance from triviality. In *First International Workshop on Parameterized and Exact Computation, IWPEC 2004, LNCS Proceedings*, volume 3162, pages 162–173. Springer.

[12] P. L. Hammer and B. Simeone. The splittance of a graph. *Combinatorica*, 1(3):275–284, 1981.

[13] P. Heggernes. Minimal triangulations of graphs: A survey. *Discrete Mathematics*, 306(3):297–317, 2006.

[14] P. Heggernes and D. Kratsch. Linear-time certifying algorithms for recognizing split graphs and related graph classes. Technical report, 2006-328, UiB, 2006.

[15] P. Heggernes and F. Mancini. Minimal split completions of graphs. In *7th Latin American Symposium on Theoretical Informatics,LATIN 2006, LNCS Proceedings*, volume 3887, pages 592–604. Springer.

[16] P. Heggernes and F. Mancini. A completely dynamic algorithm for split graphs. Technical report, 2006-334, UiB, 2006.

[17] P. Heggernes, C. Paul, J. A. Telle, and Y. Villanger. Interval completion is fixed parameter tractable. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, STOC 2007*, pages 374–381.

[18] C. W. Ho and R. C. T. Lee. Counting clique trees and computing perfect elimination schemes in parallel. *Inf. Process. Lett.*, 31(2):61–68, 1989.

[19] D. S. Johnson and C. H. Papadimitriou. On generating all maximal independent sets. *Inf. Process. Lett.*, 27(3):119–123, 1988.

[20] H. Kaplan, R. Shamir, and R. E. Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM J. Comput.*, 28(5):1906–1922, 1999.

[21] H. Kaplan, R Shamir, and R. Endre Tarjan. Tractability of parameterized completion problems on chordal and interval graphs: Minimum fill-in and physical mapping. In *35th Annual Symposium on Foundations of Computer Science, FOCS 2004*, pages 780–791. IEEE.

[22] D. Marx. Chordal deletion is fixed-parameter tractable. In *32nd International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2006, LNCS Proceedings*, volume 4271, pages 37–48. Springer.

[23] D. Marx. Parameterized coloring problems on chordal graphs. *Theor. Comput. Sci.*, 351(3):407–424, 2006.

[24] D. Marx and I. Schlotter. Obtaining a planar graph by vertex deletion. In *WG 2007, LNCS to appear.*

[25] S. D. Nikolopoulos and L. Palios. Adding an edge in a cograph. In *31st International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2005, LNCS Proceedings*, volume 3787, pages 214–226. Springer.

[26] T. Ohtsuki, L. K. Cheung, and T. Fujisawa. Minimal triangulation of a graph and optimal pivoting ordering in a sparse matrix. *J. Math. Anal. Appl.*, 54:622–633, 1976.

[27] S. Parter. The use of linear graphs in gauss elimination. *SIAM Review*, 3(2):119–130, 1961.

[28] D. G. Corneil S. Arnborg and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Algebraic and Discrete Methods*, 8:277–284, 1987.

[29] Y. Takenaga and K. Higashide. Vertex coloring of comparability+ke and -ke graphs. In *32nd International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2006, LNCS Proceedings*, volume 4271, pages 102–112. Springer.

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