REPORTS
IN
INFORMATICS

ISSN 0333-3590

Minimal split completions of graphs

Pinar Heggernes **Federico Mancini**

REPORT NO 311 **September 2005**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL http://www.ii.uib.no/publikasjoner/texrap/ps/2005-311.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

Minimal split completions of graphs*

Pinar Heggernes† Federico Mancini†

Abstract

We study the problem of adding edges to a given arbitrary graph so that the resulting graph is a split graph, called a split completion of the input graph. Our purpose is to add an inclusion minimal set of edges to obtain a minimal split completion, which means that no proper subset of the added edges is sufficient to create a split completion. Minimal completions of arbitrary graphs into chordal graphs have been studied previously, and new results have been added continuously. There is an increasing interest in minimal completion problems, and minimal completions of arbitrary graphs into interval graphs have been studied very recently. We extend these previous results to split graphs, and we give a characterization of minimal split completions, along with a linear time algorithm for computing a minimal split completion of an arbitrary input graph. Among our results is a new way of partitioning the vertices of a split graph uniquely into three subsets.

1 Introduction

Any graph can be embedded into a split graph by adding edges, and the resulting split graph is called a *split completion* of the input graph. A *minimum* split completion is a split completion with the minimum number of edges, and computing such split completions is an NP-hard problem [12]. A split completion H of a given graph G is *minimal* if no proper subgraph of H is a split completion of G. In this paper we show that a minimal split completion of a given graph can be computed in linear time.

Minimum and minimal chordal completions, also called *triangulations*, and minimum and minimal interval completions are defined analogously, by replacing split graphs with chordal graphs and with interval graphs. Computing a minimum triangulation and computing a minimum interval completion of a graph are NP-hard problems [4, 14], whereas it was shown already in 1976 that minimal triangulations can be computed in polynomial time [13]. Recently, there has been an increasing interest in minimal completion problems, which has led to faster algorithms for minimal triangulations [9, 10, 11], some of which were presented at recent years' SODA conferences, and a polynomial time algorithm for minimal interval completions [8] presented at this year's ESA conference. Minimal split completions have not been studied earlier, and with this paper we expand the knowledge about classes of graphs into which minimal completions of arbitrary graphs can be computed in polynomial time.

Minimal triangulations are well studied, and several characterizations of them have been given [7]. An algorithmically useful characterization is that a triangulation is minimal if and only if no single fill edge can be removed without destroying chordality of the triangulation [13] (fill edges are the edges added to the original graph to obtain a completion). This property does not hold for minimal interval completions. In this paper, we show that it holds for minimal split completions. Analogous to chordal graphs, we show that between a split graph $G_1 = (V, E_1)$ and a split graph $G_2 = (V, E_2)$ with $E_1 \subset E_2$, there is a sequence of split graphs that can be obtained by repeatedly removing one single edge from the previous split graph, starting from G_2. We characterize the fill edges that are candidates for removal when a non-minimal split completion H of an arbitrary graph G is given. Based on this, we give linear time algorithms both for computing minimal split completions, and for removing edges from a given split completion to obtain a minimal split completion.

*This work is supported by the Research Council of Norway through grant 166429/V30.

†Department of Informatics, University of Bergen, N-5020 Bergen, Norway. Emails: pinar@ii.uib.no and federico@ii.uib.no

This paper is organized as follows. In the next section we give the necessary graph theoretical background, assuming that the reader is familiar with the basic notions of graphs theory. In Section 3 we present our results on the split graphs sandwiched between two given split graphs, and use these results to characterize minimal split completions. A new way of partitioning the vertices of a split graph uniquely is presented in Section 4, and this is used to give a new characterization of minimal split completions in Section 5. These results are then combined to give an algorithm for removing redundant fill edges from a split completion to obtain a minimal split completion in Section 6, and an algorithm for directly computing a minimal split completion of an arbitrary input graph in Section 7. We conclude with some discussions and examples in Section 8.

2 Definitions and background

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 $d(v) = |N(v)|$. 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 $G_1 \subset G$). The complement of G is denoted by $\bar{G}$.

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 a largest clique in G, and $\alpha(G)$ to denote the size of a largest independent set in G.

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 from [6] states the possible partition configurations.

Theorem 1 (Hammer and Simeone [6]) *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).

The following theorem characterizes split graphs, and we will use condition *(iii)* to characterize minimal split completions. For this 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.

Theorem 2 (Földes and Hammer [3]) *Let G be an undirected graph. The following conditions are equivalent:*

(i) *G is a split graph.*

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

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

Remark 3 *Every induced subgraph of a split graph is also a split 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 *minimal* split completion of G if $(V, E \cup F')$ fails to be a split graph for every proper subset F' of F.

A graph is *chordal* if it contains no induced simple cycle of length at least 4. A graph is *interval* if sets of consecutive integers can be associated with its vertices such that two vertices are adjacent if and only if their associated sets intersect. (Minimal) chordal and interval completions of a given graph are defined analogously to (minimal) split completions. Chordal completions are also called *triangulations*. Both interval graphs and split graphs are chordal. For a chordal graph G, $\alpha(G)$ and $\omega(G)$ can be computed in linear time [5], whereas these are NP-hard problems for general graphs.

3 Sandwiching a split graph between two given split graphs

Given two chordal graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, such that $E_1 \subset E_2$, Rose, Tarjan, and Lueker [13] showed that there is an edge in $E_2 \setminus E_1$ whose removal from G_2 results in a chordal graph. A consequence of this result is that a given triangulation H of an arbitrary graph G is minimal if and only if it is impossible to obtain a chordal graph by removing a single fill edge from H. In this section, we show that analogous results hold for split graphs and minimal split completions. First we need the following observations.

Observation 4 *Let $G = (V, E)$ and $G' = (V, E')$ be two split graphs with $E \subseteq E'$, and let $V = I + K$ and $V = I' + K'$ be two split partitions of G and G', respectively. Then $|K' \cap K| \geq |K| - 1$.*

Proof. Assume that $|K' \cap K| < |K| - 1$. Then at least two vertices from K must be in the independent set I', but since they belong to a clique in G they must be connected in G' as well, which gives a contradiction. ∎

Observation 5 *Let $G = (V, E)$ and $G' = (V, E')$ be two split graphs with $E \subseteq E'$, and let $V = I + K$ and $V = I' + K'$ be two split partitions of G and G', respectively. Then $K' \setminus I \subseteq K$.*

Proof. We know that $V \setminus I = K$. Since $K' \subseteq V$, then $K' \setminus I \subseteq K$. ∎

Lemma 6 *Given two split graphs $G = (V, E)$ and $G' = (V, E \cup F)$ such that $E \cap F = \emptyset$, there is an edge $f \in F$ that can be removed from G' so that the result is a split graph.*

Proof. Let $V = I + K$ be a split partition of G, and let $V = I' + K'$ be a split partition of G'. If there is an edge $f \in F$ with one endpoint in I' and one endpoint in K', then f can be removed, and the resulting graph is split with split decomposition $V = I' + K'$. Assume for the rest of the proof that there is no fill edge between I' and K'.

We define the set $T = K' \cap I$, namely those vertices that belong to an independent set in the partition of G and to a clique in the partition of G'. According to our assumption, there is no fill edge between T and I'. Thus each edge in F has either both endpoints in T or is between a vertex of T and a vertex of $K' \setminus T$, since $K' \setminus T$ was already a clique in G, by Observation 5. It follows that if $F \neq \emptyset$ then $T \neq \emptyset$, and all vertices in T must be incident to some edge of F in G'. Since T is a part of an independent set in G and a part of a clique in G', there are fill edges between each pair of vertices in T. If $|T| = 1$ the fill edges connect T to $K' \setminus T$.

By Observation 4 we now have two possible situations: either $|K' \cap K| = |K|$ or $|K' \cap K| = |K| - 1$.

Assume first that $|K' \cap K| = |K|$. Then $K \subseteq K'$, and consequently $I' \subseteq I$. This means that no vertex of T is adjacent to a vertex of I' in G', because there can neither be original edges between these two sets since $T \cup I' = I$, nor edges from F since $T \subset K'$. In such a situation it is possible to pick a vertex $y \in T$ incident to one or more edges in F, and remove any of the fill edges incident to y. Doing this the graph will remain split because we can still partition it in an independent set $I' \cup \{y\}$ and a clique $K' \setminus \{y\}$. We proved above that such vertex y must exist.

Let now $|K' \cap K| = |K| - 1$. Then there must be a vertex x that in G belongs to K and in G' belongs to I', such that $(I' \setminus \{x\}) \subseteq I$. Now, each vertex of T can be adjacent to at most one vertex of I', namely x. If there is at least one vertex $y \in T$ which is not adjacent to x, then we can proceed as the previous case. If all vertices of T are adjacent to x, then $N(x) = K'$, so we can just swap x with any vertex in $y \in T$ incident to an edge of F, and remove this edge, since it now connects the independent set to the clique. Swapping the vertices we make a new partition where x is in the clique and y in the independent set, and thus the result is a split graph.

Note that if $F = \emptyset$ then it means that $G = G'$ and either $I = I'$ and $K = K'$, or $I + K$ and $I' + K'$ are two possible partitions of the same split graph G. ∎

Corollary 7 *Given two split graphs $G = (V, E)$ and $G' = (V, E \cup F)$ with $E \cap F = \emptyset$, there is a sequence of split graphs G_0, G_1, G_2, ..., $G_{|F|}$ such that G_{i-1} is obtained by removing edge f_i from G_i, for $1 \leq i \leq |F|$, where $G_0 = G$, $G_{|F|} = G'$, and $F = \{f_1, f_2, ..., f_{|F|}\}$.*

Theorem 8 *Given an arbitrary graph G and a split completion G' of G, G' is a minimal split completion if and only if no single fill edge can be removed from G' without destroying the split property.*

Proof. If G' is a minimal split completion then no subset of its fill edges can be removed, so no single fill edge can be removed either. If G' is not a minimal split completion, another split graph G'' exists between G and G'. Then by Lemma 6, there is a single fill edge that can be removed from G' while preserving the split property. ∎

Thus we have a characterization of minimal split completions. We will use this to give another characterizations of minimal split completions and to describe the fill edges that can be removed from non-minimal split completions in Section 5. First, in the next section, we define a new way of partitioning the vertices of a split graph uniquely.

4 Unique decompositions of split graphs

In this section, as an alternative to split partitions, we define another way of partitioning the vertices of a split graph that will be useful to decide whether a given split completion is minimal or not. We will call the new partition a *decomposition.*

When we are given a non-minimal split completion of an arbitrary graph, according to Lemma 6, the redundant fill edges can be removed one by one until we reach a minimal split completion. The edges that can be removed without problems are the ones connecting the independent set with the clique in the split partition of the completion. However, since this partition is not necessarily unique, and since we do not know the underlying minimal split completion, problems occur according to cases *(ii)* and *(iii)* of Theorem 1. To avoid this ambiguity we define a third set of vertices in the graph, that we will call Q. In case we do not have a unique split partition, this set will contain those vertices that can be chosen to be either in the independent set or in the clique, determining different partitions.

Definition 9 *Given a split graph $G = (V, E)$ that has no unique split partition, we define a* decomposition *$V = S + C + Q$ of G as follows:*

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

If G has a unique split partition $V = I + K$, we do not need such a decomposition, but for completeness, we define $S = I$, $C = K$, and $Q = \emptyset$ in this case, so that a decomposition is always defined. (Note that there can be vertices of degree $\omega(G) - 1$ in G also when its split partition is unique.) For a split graph G, $\omega(G)$, $\alpha(G)$, and the corresponding maximum clique and independent set can be computed in linear time [5]. Thus it can be decided by Theorem 1 whether G has a unique split partition or not. Hence the decomposition of a split graph is uniquely defined. Figure 1 shows examples of decompositions of some split graphs.

Lemma 10 *Let $G = (V, E)$ be a split graph with no unique split partition, and let $V = S + C + Q$ be the decomposition of G. Then*

(i) $S \subseteq I$ *and* $C \subseteq K$, *for every split partition* $V = I + K$ *of* G.

(ii) $Q \neq \emptyset$.

(ii) Q *is exactly the set of vertices each of which belongs to a clique and to an independent set in two different split partitions of* G, *respectively.*

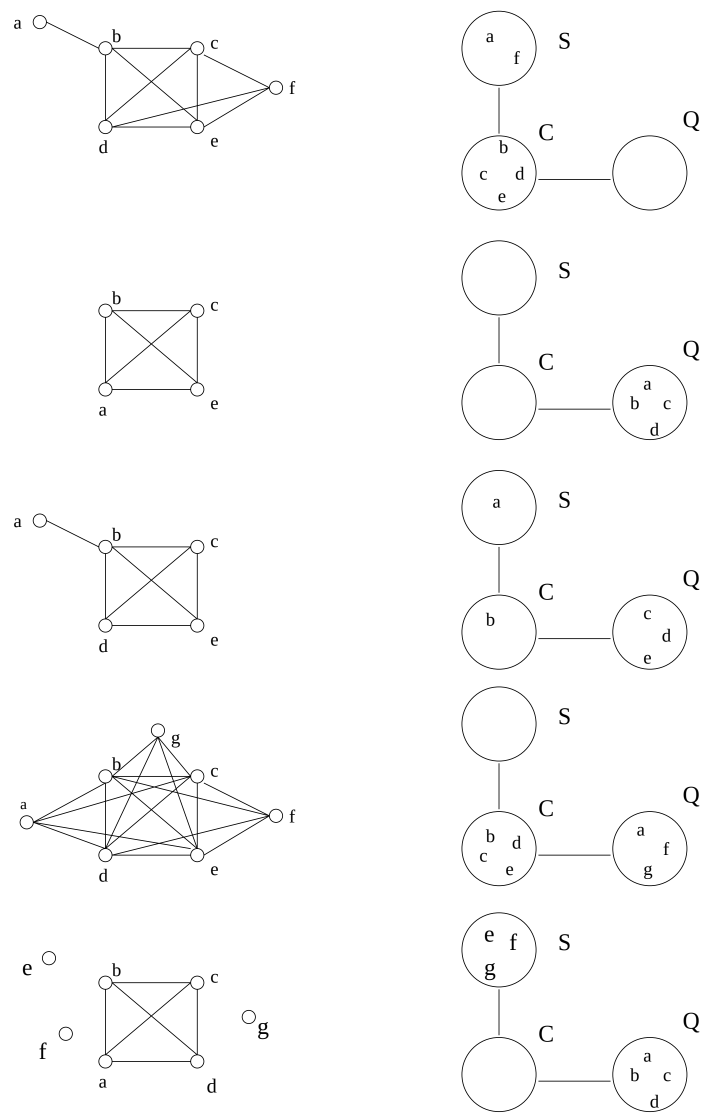


Figure 1: *Examples of decompositions for different kind of split graphs.*

Proof.

(i) Let $V = I + K$ be any split partition of G. By Theorem 1, each vertex of K belongs also to a clique of maximum size, and thus has degree at least $\omega(G) - 1$. Therefore, a vertex that has degree less than $\omega(G) - 1$ cannot belong to K, and it must belong to I. A vertex of I can be adjacent to at most $\omega(G) - 1$ vertices, because otherwise we have a clique of size $\omega(G) + 1$. Thus, a vertex that has degree more than $\omega(G) - 1$ must belong to K.

(ii) Let $V = I + K$ be any split partition of G. By Theorem 1, either there is a vertex x in I such that $K \cup \{x\}$ is a clique, or there is a vertex y in K such that $I \cup \{y\}$ is an independent set. In either case, each such vertex x or y is adjacent to all vertices of K and to no other vertex, and by Theorem 1, it has degree exactly $\omega(G) - 1$. Thus $Q \neq \emptyset$.

(iii) By the argument in *(ii)* every vertex that can be moved between an independent set and a clique in some split partition of G must have degree $\omega(G) - 1$. Let us show that each vertex of degree $\omega(G) - 1$ can indeed be moved between partitions. Let $V = I + K$ be any split partition of G, and let v be a vertex of degree $\omega(G) - 1$. Assume first that $v \in I$. If $|K| = \omega(G) - 1$, then by moving v from I to K, we get another split partition of G. If $|K| = \omega(G)$, then we know by Theorem 1 that a vertex x of K can be moved to I to give a different split partition. Thus x cannot be adjacent to v. We can swap v and x between I and K, and get a new split partition. Assume now that $v \in K$. If $|K| = \omega(G) - 1$, then there must be a vertex z in I that is adjacent to all vertices of K. Thus z is the only neighbor of v outside of K. We can swap z and v and get another partition. If $|K| = \omega(G)$ then v has no neighbors in I and we can move v from K to I and get a new partition. ∎

Corollary 11 *A split graph $G = (V, E)$ has a unique partition $V = I + K$ if and only if there are exactly $\omega(G)$ vertices of degree $> \omega(G) - 1$.*

Corollary 12 *Let $G = (V, E)$ be a split graph with decomposition $V = S + C + Q$. Then every vertex of Q is adjacent to all vertices of C and to no vertex of S.*

Lemma 13 *Let $G = (V, E)$ be a split graph with decomposition $V = S + C + Q$. Then one of the following is true:*

(i) Q is a clique and $|C| + |Q| = \omega(G)$.
(ii) Q is an independent set, $|C| = \omega(G) - 1$, and $|Q| \geq 2$.

Proof. If Q is empty, there is nothing to prove. Assume that $|Q| \geq 1$, so that the split partition of G is not unique. If $\omega(G) > 1$ (otherwise the graph is a set of disconnected vertices and they would be all in Q) then $|C| + |Q| \geq \omega(G)$, so we can distinguish two situations: $|C| + |Q| = \omega(G)$ or $|C| + |Q| > \omega(G)$.

If $|C| + |Q| = \omega(G)$ then $|Q|$ is a clique, because the largest clique in G must have size $\omega(G)$ and it can only be obtained by adding to C all vertices of Q.

If $|C| + |Q| > \omega(G)$ then we will show that $|C| = \omega(G) - 1$ and Q is an independent set. If $|C| < \omega(G) - 1$ then $|Q| > 2$, and there must be at least a subset $Q' \subset Q$ that is a clique of size $\omega(G) - |C| \geq 2$. All vertices of Q' have degree $\omega(G) - 1$, and since they make a clique of size $\omega(G)$ with C, they cannot be adjacent to any vertex in $Q \setminus Q'$. The vertices in $Q \setminus Q'$ must be an independent set, or it would not be possible to make a split partition $V = (C \cup Q') + (S \cup (Q \setminus Q'))$, so they are adjacent only to C (by Corollary 12) and consequently each of them has degree at most $\omega(G) - 2$, contradicting the fact that they belong to Q. So we must have $|C| = \omega(G) - 1$ and $|Q| \geq 2$, which implies that Q is an independent set, since otherwise we would get a too large clique in G by Corollary 12. ∎

Corollary 14 *Let $G = (V, E)$ be a split graph with decomposition $V = S + C + Q$ and $Q \neq \emptyset$. Then in any split partition $V = I + K$ of G, at least $\omega(G) - 1 - |C|$ vertices of Q belong to K.*

Lemma 15 *Let $G = (V, E)$ be a split graph with decomposition $V = S + C + Q$ and $q \neq \emptyset$. If $|Q| = 1$, then $|C| = \omega(G) - 1$ and $|S| \geq 2$. If $|C| + |Q| = \omega(G)$ and $|Q| > 1$, then $|S| \geq 1$.*

Proof. Every vertex of C has degree greater than $\omega(G)-1$, so every vertex of C has at least one neighbor in S (since $|C|+|Q| = \omega(G)$), but every vertex of S has degree at most $\omega(G)-2$, so at least two of them are needed to be connected to C in the first case, and at least one in the last. ∎

In the next section, we will use these results to characterize minimal split completions of arbitrary graphs.

5 Characterizing minimal split completions

Assume that we are given an arbitrary graph $G = (V, E)$ and a split completion $H = (V, E \cup F)$ of G. We want to find a sufficient and necessary condition for H to be a minimal split completion of G. First we identify the fill edges that can be removed from any non-minimal split completion. Note that, when $V = S + C + Q$ is the decomposition of H, any fill edge is either incident to a vertex of $S \cup Q$ or both of its endpoints belongs to C.

Lemma 16 *Let $H = (V, E + F)$ be a split completion of an arbitrary graph $G = (V, E)$, and let $V = S + C + Q$ be the decomposition of H. Then any fill edge incident to a vertex in $S \cup Q$ can be removed so that the resulting graph is split.*

Proof. We will prove that there is a split partition $V = I + K$ of H such that any fill edge incident to a vertex of $S \cup Q$ has one endpoint in I and one endpoint in K, and we know that such edges can be removed. We also know that all vertices of S belong to the independent set of any split partition of H, and all vertices of C belong to the clique of any split partition of H. This means that any edge between S and C can be removed. Let us then assume that there are no fill edges connecting S and C. Remember also that there are no edges between S and Q. Let us also assume that the partition is not unique, so that $Q \neq \emptyset$. Under these assumptions, each fill edge incident to a vertex of $S \cup Q$ can only be between two vertices of Q or between a vertex of Q and a vertex of C, and we have the following cases.

Case 1: Q is a clique and there is a fill edge between two vertices $x, y \in Q$. If Q is a clique, then we can make a partition where at most one of the vertices of Q is chosen to be in the independent set of a partition and all the others must be in the clique. If we put x (or y) in the independent set and y (or x) in the clique, there will be a fill edge (xy) between the independent set and the clique, that we can remove.

Case 2: There is a fill edge between Q and C. If a fill edge is between a vertex $x \in Q$ and a vertex $y \in C$, since we can always choose at least one vertex of Q to be in the independent set of a partition regardless of whether Q is a clique or an independent set, let us choose exactly x. Since y is in C it will always be in the clique of any partition of H, so we now have a fill edge connecting a vertex of the independent set (x) to a vertex of the clique (y), and we can remove it. ∎

Note that Lemma 16 does not mean that all fill edges incident to $S \cup Q$ can be removed. We are guaranteed to be able to remove one such edge. After that the decomposition of the resulting graph might change, and thus the set of fill edges that can be removed might also change. In the next section, we will describe precisely how the sets S, C, and Q might change after removing an unnecessary fill edge.

Lemma 17 *Let $H = (V, E + F)$ be a split completion of an arbitrary graph $G = (V, E)$, and let $V = S + C + Q$ be the decomposition of H. If each fill edge has both its endpoints in C, then H is a minimal split completion of G.*

Proof. Assume that G, H, S, C, and Q are as in the premise of the lemma such that all fill edges of H have both their endpoints in C. Thus if $F \neq \emptyset$ then $|C| \geq 2$ and $\omega(H) \geq 2$. We show that removing any single fill edge from H results in a non-split graph.

If Q is empty and thus H has a unique split partition, then by Theorem 1 and Corollary 11, $|C| = \omega(H)$, no vertex of S is adjacent to whole C, and every vertex of C has a neighbor in S. Hence $|S| \geq 2$. If $|C| = 2$ then the single edge in C is a fill edge. Removing it we would get a $2K_2$, because we can pick two vertices in S adjacent each to only one vertex of C. If $|C| > 2$ then removing a fill edge we get two nonadjacent vertices $x, y \in C$. Now, x and y must each have a neighbor in S. If they have a common neighbor w, then we can find a vertex $v \in C$ which is not adjacent to w, since no vertex of S is adjacent to every vertex of C. But x and y are both adjacent to v, so this results in an induced cycle w, x, v, y, w of length 4. If they do not have a common neighbor, then there exist $w, z \in S$, where w is adjacent to x and not to y, and z is adjacent to y and not to x, So removing the edge between x and y we get a $2K_2$.

If $Q \neq \emptyset$ then S can be even empty or disconnected from C. Let us work on Q and C using Lemma 13 and 15.

In the case when $|C| + |Q| = \omega(H)$, we have that Q is a clique and $S \neq \emptyset$. If $|Q| = 1$, then $|C| = \omega(H) - 1$, so there must be at least 2 vertices in S adjacent to vertices of C, and no vertex of S is adjacent to every vertex of C, so we can use the same argument as above. If $|Q| > 1$, then $|C| < \omega(H) - 1$. Thus every vertex of C has a neighbor in S, and either we have the previous case, or there is a vertex $z \in S$ adjacent to all vertices in C. Recall that every vertex of Q is connected to all vertices of C and to no vertex of S. Let us now take any $x, y \in C$ and remove the edge xy. We can find a vertex $w \in Q$, adjacent to both x and y so that the subgraph induced by $\{z, x, w, y\}$ is a cycle of length 4.

In the case when $|C| + |Q| > \omega(H)$, we have that Q is an independent set, and $|C| = \omega(H) - 1$. In this case S can be empty. However, since $|C| \geq 2$, then $\omega(H) \geq 3$ and $|Q| \geq 2$. Since every vertex of Q is adjacent to all vertices of C, we can find two vertices w and z in Q, such that if we remove any fill edge xy from C, we get an induced cycle w, x, z, y, w of length 4. ■

Theorem 18 *Let $H = (V, E + F)$ be a split completion of an arbitrary graph $G = (V, E)$, and let $V = S + C + Q$ be the decomposition of H. H is a minimal split completion of G if and only if all fill edges have both endpoints in C.*

Proof. One direction follows from Lemma 17. For the other direction assume that H is minimal. Then no single fill edge can be removed without destroying split property. Given the decomposition $V = S+C+Q$, then each fill edge can have one endpoint in $S \cup Q$ and one endpoint in C, or both endpoints in C or in Q. This is because there cannot be fill edges between S and Q (by Corollary 12), and within S (it is an independent set). By Lemma 16 a fill edge incident to vertices in $S \cup Q$ can always be removed, so since the completion is minimal, the only possible fill edges are the ones in C. ■

6 Obtaining a minimal split completion from a given split completion

In the next section we will give an algorithm that computes a minimal split completion of any given graph. However, for some applications it might be desirable to compute a minimal split completion of that fits within an already given split completion. This problem has been studied and solved for minimal triangulations [1, 2], and we solve it for split completions in this section.

Assume that we are given an arbitrary graph G, a split completion H of G, and the decomposition $S + C + Q$ of H. By the results of the previous section, we know how to decide whether H is a minimal split completion, and if not, we know that any single fill edge incident to $S \cup Q$ can be removed. After this removal, the sets S, C, and Q might change. So a straight forward algorithm to remove redundant fill edges from H to obtain a minimal split completion $M \subseteq H$ of G, would be to remove a fill edge incident to $S \cup Q$, recompute the decomposition of the resulting split completion, and continue until a minimal split completion is reached. In this section, we will show that a new decomposition of the intermediate graph does not have to be recomputed from scratch, and that a minimal split completion can be reached in time linear in the size of H.

Theorem 19 *Let $H = (V, E \cup F)$ be a split completion of an arbitrary graph $G = (V, E)$, with $F \cap E = \emptyset$. A minimal split completion M of G, such that $G \subseteq M \subseteq H$, can be computed in time $O(|V| + |E| + |F|)$.*

Proof. Let $V = S + C + Q$ be the decomposition of H. Let pq be a fill edge that can be removed, H' the graph that is the result of removing pq from H, and $V = S' + C' + Q'$ the decomposition of H'. We know that pq can be of three types. Based on this, for each fill edge that can be removed, we have to analyze all possible cases, and show that in each case, S', C', and Q' can be computed from S, C, and Q in amortized constant time. Since we have a constant number of cases to check for each fill edge that can be removed, and since we can check in constant time whether a particular fill edge can be removed by the results of Section 5, the total number of steps will be at most F, and the total work we do will be linear in the size of H. To ease the notation, we let $w = \omega(H)$ and $w' = \omega(H')$.

Case 1: $p \in S$ and $q \in C$.

Case 1.1: Vertex q has degree w in H.
Then $S' = S$, $C' = C \setminus \{q\}$, and $Q' = Q \cup \{q\}$.
The partition of H can be either unique or not, but H' will for sure not have a unique partition. In fact removing pq, the degree of q will become $w - 1$, so that it cannot belong to C anymore, and since $w = w'$, in H', $|C|$ will not be equal to w' (as required by Corollary 11 to have a unique partition). It follows that q must be moved to Q.

Case 1.2: Vertex q has degree more than w in H.
Then $S' = S$, $C' = C$, and $Q' = Q$.
Since the $w = w'$ and no vertex of C gets a degree smaller than w, either H had a unique partition or not, H' will still have the same kind of partition as H.

Case 2: $p \in Q$ and $q \in Q$.

Case 2.1: $|Q| = 2$.
Then $S' = S \setminus W$, $C' = C$, $Q' = Q \cup W$, where $W = \{v \in S \mid d_{H'}(v) = w - 2\}$.
Before the removal of edge pq, $d(q) = d(p) = w - 1$, and after the removal we have $d(q) = d(p) = w - 2 = w' - 1$. Removing pq, the size of the maximum clique becomes $w' = w - 1$, so all vertices whose degree has become $w - 2 = w' - 1$ must be moved to Q.

Case 2.2: $|Q| > 2$.
Then $S' = S$, $C' = C \cup Q \setminus \{p, q\}$, and $Q' = \{p, q\}$.
Removing pq, the size of the maximum clique becomes $w' = w - 1$ as the previous case, but since $|C| < w - 2$, no vertex in S gets degree $w - 2$. Vertices p and q stay in Q for the same reason as before, but all the other vertices in Q still have degree $w - 1 = w'$, so they must be moved to C.

Case 3: $p \in Q$ and $q \in C$.

Case 3.1: $|Q| = 1$.
Then $S' = S \cup \{p\}$, $C' = C$, and $Q' = \emptyset$.
Removing pq, the size of the maximum clique becomes $w' = w - 1$, and $|C| = w - 1 = w'$, so the new graph has a unique partition that we construct as defined in Section 4.

Case 3.2: $|Q| = 2$.

Case 3.2.1: Vertex q has no neighbors in S.
Then $S' = S \cup \{p\}$, $C' = C \setminus \{q\}$, and $Q' = Q \cup \{q\} \setminus \{p\}$.
If q has no neighbors in S, then its degree is w in H and becomes $w - 1$ in H', and it is moved to Q. In this case, Q is an independent set, the size of the maximum clique is not affected, so $d(p) = w - 2$ in H', and p must be moved to S.

Case 3.2.2: Vertex q has neighbors in S and Q is an independent set.
Then $S' = S \cup \{p\}$, $C' = C$, and $Q' = Q \setminus \{q\}$.
In this case the degree of $q > w - 1$ even after we delete the fill edge, so we only need to move p to S.

Case 3.2.3: Vertex q has neighbors in S and Q is a clique.
Then the same thing happens as in Case 2.1, but also the single neighbor of p in Q must be moved to C.

Case 3.3: $|Q| > 2$.

Case 3.3.1: Q is an independent set.
Then $S' = S \cup \{p\}$, $C' = C$, and $Q' = Q \setminus \{p\}$.
Like in the previous case, q has enough neighbors to guarantee that its degree will not decrease to $w - 1$, so we only need to move p to S.

Case 3.3.2: Q is a clique.
Then $S' = S$, $C' = C \cup Q \setminus \{p\}$, and $Q' = \{p\}$.
The argument is the same as in Case 2.2.

Each of the conditions can be checked in constant time for each edge pq. Most operations require constant time, and some operations require moving several vertices from S to Q, and from Q to C. However, elements are always moved one by one into S and out of C. Thus an algorithm that removes all redundant fill edges from H is obtained directly from the cases above, and an amortized analysis gives running time $O(|V| + |E| + |F|)$. ■

7 Computing a minimal split completion directly

In this section, we show that minimal split completion of a given graph can be computed in time linear in the size of the input graph. An simple and intuitive method to embed an arbitrary graph $G = (V, E)$ into a split graph, is to select a maximal independent set I of G and add edges to make $V \setminus I$ into a clique. This procedure does not guarantee that the resulting split completion is minimal, since it can add edges even to a graph that is already split, in particular if the graph does not have a unique partition, as illustrated in Figure 2. However, it can be modified to compute a minimal split completion, by choosing vertices of minimum degree first when computing the maximal independent set. We call this modified algorithm *MinimalSplit* and present it below.

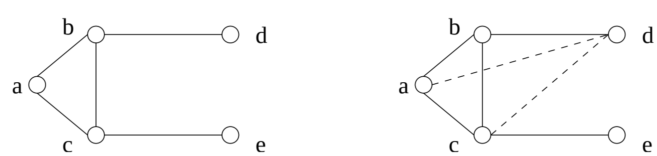


Figure 2: *The graph on the left is already split, but choosing the set $M = \{b, e\}$ as the maximal independent set and making the rest a clique, we get a larger split graph.*

Algorithm MinimalSplit
Input: An arbitrary graph $G = (V, E)$;
Output: A minimal split completion $H = (V, E \cup F)$ of G.
$I = \emptyset$; $K = \emptyset$;
Unmark all vertices;
while there are unmarked vertices in V **do**
Choose an unmarked vertex v with minimum degree in G;
Mark and add v to I;
Mark and add all neigbors of v to K;
end-while
Make K into a clique adding a set F of fill edges;
$H = (V, E \cup F)$;

Lemma 20 *Given an arbitrary graph $G = (V, E)$, the graph $H = (V, E \cup F)$ computed by Algorithm MinimalSplit is a minimal split completion of G.*

Proof. Let $V = I + K$ be the split partition of H computed by the algorithm. By construction, I is an independent set, K is a clique, and no edges are added between I and K, so H is a split graph. It follows from Lemma 17 that if $V = I + K$ is a unique partition of H, then H is a minimal split completion.

Let us consider the case when $V = I + K$ is not a unique partition of H, and let $V = S + C + Q$ be the decomposition of H. If $|K| < 2$ then no edges are added by the algorithm, so the completion is trivially minimal. Assume therefore that $|K| \geq 2$. By construction, $K = \bigcup_{v \in I} N(v)$, so every vertex of K has a neighbor in I. It follows that $|K| = \omega(H) - 1$. Otherwise (if $|K|$ were $\omega(H)$), there would be $\omega(H)$ vertices in K with degree greater then $\omega(H) - 1$, contradicting Corollary 11. Since the split partition is not unique, there is at least one vertex z of degree $\omega(H) - 1$ in I, that can be moved to K by Theorem 1. Such vertices z belong to Q, but they are not adjacent to any fill edge. Consequently, the only possibility for the completion to be non-minimal is that a vertex $x \in K$ incident to a fill edge, has degree $\omega(H) - 1$ so that x belongs to Q. Thus x has exactly one neighbor in I. This means that there is exactly one vertex $y \in I$ of degree $\omega(H) - 1$, since vertices of degree $\omega(H) - 1$ in I must be adjacent to all vertices of K. So we have exactly one vertex $y \in I$ of degree $\omega(H) - 1$ and a vertex $x \in K$ of degree $\omega(H) - 1$, such that $N(x) \cap I = \{y\}$. The degree of x in the input graph G is actually less than $\omega(H) - 1$, because it is incident to at least one fill edge. But the degree of y is $\omega(G) - 1$ also in G since no edges are added to vertices in I. This means that $d(x) < d(y)$ in G, but a vertex can be in K only if one of its neighbors in G has been selected before it to be in I. In this case, since the only neighbor of x in G selected to be in I is y, it means that y has been processed by the algorithm before x, but that is a contradiction because $d(x) < d(y)$, and the algorithm always chooses the vertex with minimum degree among the unprocessed ones.

This means that any graph obtained by the algorithm is a minimal split completion of the input graph by Lemma 17. ■

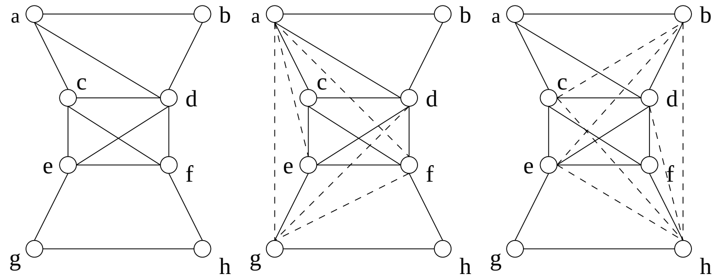
Let us consider the time complexity of this algorithm. Since we add edges only between the vertices of K, we can actually skip the step of adding edges, because the resulting split partition will uniquely define the edges of H. Thus the algorithm can be modified to return just I and K (the edges between I and K are all edges of G). The degrees are computed only in the beginning of the algorithm, and need not be recomputed. This and the rest of the algorithm clearly require at most $\sum_{v \in V} d(v)$ steps, which sums up to time $O(|V| + |E|)$.

8 Conclusions

We have given a characterization of minimal split completions and we have shown how to compute minimal split completions in linear time. We have also given an algorithm for computing a minimal split completion between the input graph G and an already given non-minimal split completion H of G. To achieve these goals, we introduced a new way of uniquely partitioning the vertices of a split graph into three subsets instead of two.

With these results, polynomial time algorithms are now known for minimal triangulations, minimal interval completions, and minimal split completions of arbitrary graphs. There are other interesting graph classes into which any graph can be embedded by adding edges. We are interested in knowing whether minimal completions into these classes can be computed in polynomial time. Also, is there a graph class $\mathcal{C}$ such that minimal $\mathcal{C}$ completion of arbitrary graphs is an NP-hard problem?

Finally, we would like to mention that even if we were able to pick a maximum independent set of the arbitrary input graph G, we are not guaranteed to get a minimum split completion by completing the rest of the vertices into a clique, as illustrated in Figure 3. We suspect that one always gets a minimal split completion with this approach, but this will in any case not give a practical algorithm for general graphs, since finding a maximum independent set is an NP-hard problem.



(a) Inputgraph (b) Completion1 (c) Completion2

Figure 3: *Given the input graph $G = (V, E)$, we can find at least two different maximum independent sets: $M_1 = \{b, c, h\}$ and $M_2\{a, f, g\}$. In fig. 3(b) we make $V \setminus M_1$ in a clique adding 5 edges; In fig. 3(c) we make $V \setminus M_2$ in a clique adding 6 edges. This means that even taking a maximum independent set, we cannot be sure to get a minimum completion.*

Acknowledgment

We thank Jan Arne Telle for an initial discussion on the example given in Figure 2.

References

[1] A. Berry, J-P. Bordat, P. Heggernes, G. Simonet, and Y. Villanger. A wide-range algorithm for minimal triangulation from an arbitrary ordering. *J. Algorithms*. To appear.

[2] J. R. S. Blair, P. Heggernes, and J. A. Telle. A practical algorithm for making filled graphs minimal. *Theor. Comput. Sci.*, 250:125–141, 2001.

[3] S. Földes and P. L. Hammer. Split graphs. *Congressus Numerantium*, 19:311–315, 1977.

[4] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman and Co., 1978.

[5] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, 1980.

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

[7] P. Heggernes. Minimal triangulations of graphs - a survey. *Dicrete Mathematics*. To appear.

[8] P. Heggernes, K. Suchan, I. Todinca, and Y. Villanger. Minimal interval completions. In *Algorithms - ESA 2005*. Springer Verlag, 2005. LNCS to appear.

[9] P. Heggernes, J. A. Telle, and Y. Villanger. Computing minimal triangulations in time $O(n^{\alpha} \log n) = o(n^{2.376})$. In *Proceedings of SODA 2005 - 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 907–916, 2005.

[10] D. Kratsch and J. Spinrad. Between $O(nm)$ and $O(n^{\alpha})$. In *Proceedings of SODA 2003 - 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 709–716, 2003.

[11] D. Kratsch and J. Spinrad. Minimal fill in $o(n^3)$ time. 2004. Submitted.

[12] A. Natanzon, R. Shamir, and R. Sharan. Complexity classification of some edge modification problems. *Disc. Appl. Math.*, 113:109–128, 2001.

[13] D. Rose, R.E. Tarjan, and G. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5:146–160, 1976.

[14] M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Alg. Disc. Meth.*, 2:77–79, 1981.