Laboratoire de l'Informatique du Parallélisme

École Normale Supérieure de Lyon
Unité Mixte de Recherche CNRS-INRIA-ENS LYON n° 5668

Listing All Potential Maximal Cliques of a Graph

Vincent Bouchitté
Ioan Todinca

September 1999

Research Report N° 99-40

École Normale Supérieure de Lyon
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : +33(0)4.72.72.80.37
Télécopieur : +33(0)4.72.72.80.80
Adresse électronique : `lip@ens-lyon.fr`

Listing All Potential Maximal Cliques of a Graph

Vincent Bouchitté
Ioan Todinca

September 1999

Abstract

A potential maximal clique of a graph is a vertex set that induces a maximal clique in some minimal triangulation of that graph. It is known that if these objects can be listed in polynomial time for a class of graphs, the treewidth and the minimum fill-in are polynomially tractable for these graphs. We show here that the potential maximal cliques of a graph can be generated in polynomial time in the number of minimal separators of the graph. Thus, the treewidth and the minimum fill-in are polynomially tractable for all graphs with polynomial number of minimal separators.

Keywords: treewidth, minimum fill-in, minimal separators, potential maximal cliques

Résumé

Une clique maximale potentielle d'un graphe est un ensemble de sommets qui induit une clique maximale dans au moins une triangulation minimale de ce graphe. Il a été prouvé que si ces objets peuvent être énumérés en temps polynomial pour une classe de graphes, la largeur arborescente et la complétion minimale sont calculables en temps polynomial pour ces graphes. Nous montrons ici que les cliques maximales potentielles d'un graphe peuvent être générées en temps polynomial par rapport au nombre de ses séparateurs minimaux. En conséquence, la largeur arborescente et la complétion minimale sont calculables en temps polynomial pour tous les graphes ayant un nombre polynomial de séparateurs minimaux.

Mots-clés: largeur arborescente, complétion minimale, séparateurs minimaux, cliques maximales potentielles

1 Introduction

The notion of *treewidth* was introduced at the beginning of the eighties by Robertson and Seymour [25, 26] in the framework of their graph minor theory. A graph H is a minor of a graph G if we can obtain H from G by using the following operations: discard a vertex, discard an edge, merge the endpoints of an edge in a single vertex. Among the deep results obtained by Robertson and Seymour, we can cite this one: every class of graphs closed by minoration which does not contain all the planar graphs has bounded treewidth.

A graph is *chordal* or *triangulated* if every cycle of length greater or equal to four has a chord, i.e. edge between two non-consecutive vertices of the cycle. A triangulation of a graph is a chordal embedding, that is a supergraph, on the same vertex set, which is triangulated. The treewidth problem is to find a triangulation such that the size of the biggest clique is as small as possible. Another closed problem is the minimum fill-in problem. Here we have to find a triangulation of the graph such that the number of the added edges is minimum. In both cases we can restrict to *minimal triangulations*, i.e. triangulations with a set of edges minimal by inclusion.

The treewidth and the minimum fill-in play an important role in various areas of computer science e.g. sparse matrix factorization [27], and algorithmic graph theory [3, 14, 2, 8]. For an extensive survey of these applications see also [5, 7].

Computing the treewidth is equivalent to find a tree decomposition, that is a tree such that each node of the tree is labeled by a vertex set of the graph. The labels of the nodes must respect some constraints: every vertex of the graph must appear in some label, the endpoints of an edge must appear in a same label, if a same vertex is in two different labels it must be in all the labels on the unique path of the tree connecting the two occurrences of the vertex. The width of the tree decomposition is then the size of the largest label minus one, and the treewidth is the smallest width over all the tree decompositions of the graph. Many graph problems that model real-life problems are intractable in the sense that they are NP-hard. If we deal with a class of graphs of bounded treewidth most of these problems become polynomial and even linear e.g. maximum independent set, hamiltonian circuit or Steiner tree. There are two ways to solve problems when the treewidth is bounded, the first uses dynamic programming [5, 16] and the second is based upon reduction techniques [2, 8].

Unfortunately the computation of the treewidth and of the minimum fill-in of a graph are NP-hard [1, 30] even for co-bipartite graphs. However, a polynomial time approximation algorithm with $\mathcal{O}(\log n)$ performance ratio is described in [9]. The problem of the existence of a polynomial approximation of the treewidth within a multiplicative constant remains still open. For any fixed constant k, there exist polynomial algorithms finding a tree decomposition of width at most k if such a decomposition exists. Arnborg et al. [1] gave the first algorithm that solves this problem in $\mathcal{O}(n^{k+2})$ time. Since numerous improvements have been done on the domain until the linear time algorithm of Bodlaender [6]. Notice that the constant hidden by the $\mathcal{O}$ notation is doubly exponential in k^2. Some results for treewidth appeared in the literature in connection with logic. The works by Arnborg et al. [2], Courcelle [13], Courcelle and Mosbah [14] led to the conclusion that all the problems which are expressible in extended monadic second order logic can be solved in linear time for graphs of bounded treewidth.

There exist several classes of graphs with unbounded treewidth for which we can solve polynomially the problem of the treewidth and the minimum fill-in. Among them there are the chordal bipartite graphs [19, 12], circle and circular-arc graphs [28, 23], AT-free graphs with polynomial number of minimal separators [22]. Most of these algorithms use the fact that these classes of graphs have a polynomial number of *minimal separators*. It was conjectured in [17, 18] that the treewidth and the minimum fill-in should be tractable in polynomial time for all the graphs having a polynomial number of minimal separators. We solve here this ESA'93 conjecture.

The crucial interplay between the minimal separators of a graph and the minimal triangulations was pointed out by Kloks, Kratsch and Müller in [21], these results were concluded in Parra and Scheffler [24]. Two minimal separators S and T cross if T intersects two connected components of $G\backslash S$, otherwise they are parallel. The result of [24] states that a minimal triangulation is obtained by considering a maximal set of pairwise parallel separators and by completing them i.e. by adding all the missing edges inside each separator. However this characterization gives no algorithmic information about how we should construct a minimal triangulation in order to minimize the cliquesize or the fill-in.

Trying to solve this later conjecture, we studied in [10, 11] the notion of *potential maximal clique*. A vertex set K is a potential maximal clique if it appears as a maximal clique in some minimal triangulation. In [10], we characterized a potential maximal clique in terms of the *maximal sets of neighbor separators*, which are the minimal separators contained in it. We designed an algorithm which takes as input the graph and the maximal sets of neighbor separators and which computes the treewidth in polynomial time in the size of the input. For all the classes mentioned above we can list the maximal sets of neighbor separators in polynomial time, so we unified all the previous algorithms. Actually, the previous algorithms compute the maximal sets of neighbor separators in an implicit manner. In [11], we gave a new characterization of the potential maximal cliques avoiding the minimal separators. This allowed us to design a new algorithm that, given a graph and its potential maximal cliques, computes the treewidth and the minimum fill-in in polynomial time. Moreover this approach permitted us to solve the two problems for a new class of graphs, namely the weakly triangulated graphs. It was probably the last natural class of graphs with polynomial number of minimal separators for which the two problems remained open.

This paper is devoted to solve the ESA'93 conjecture, that is the treewidth and the minimum fill-in are polynomially tractable for the whole class of graphs having a polynomial number of minimal separators. Recall that if we are able to generate all the potential maximal cliques of any graph in polynomial time in the number of its minimal separators, then the treewidth and the minimum fill-in are also computable in polynomial time in the number of minimal separators. We define the notion of *active separator* for a potential maximal clique which leads to two results. First, the number of potential maximal cliques is polynomially bounded by the number of minimal separators. Secondly, we are able to enumerate the potential maximal cliques in polynomial time in their number. These results reinforce our conviction that the potential maximal cliques are the pertinent objects to study when dealing with treewidth and minimum fill-in.

2 Preliminaries

Throughout this paper we consider finite, simple, undirected and connected graphs.

Let $G = (V, E)$ be a graph. We will denote by n and m the number of vertices, respectively the number of edges of G. For a vertex set $V' \subseteq V$ of G, we denote by $N_G(V')$ the neighborhood of V' in $G\backslash V'$ – so $N_G(V') \subseteq V\backslash V'$.

A subset $S \subseteq V$ is an a, b*-separator* for two nonadjacent vertices $a, b \in V$ if the removal of S from the graph separates a and b in different connected components. S is a *minimal* a, b*-separator* if no proper subset of S separates a and b. We say that S is a *minimal separator* of G if there are two vertices a and b such that S is a minimal a, b-separator. Notice that a minimal separator can be strictly included in another one. We denote by Δ_G the set of all minimal separators of G.

Let G be a graph and S a minimal separator of G. We note $\mathcal{C}_G(S)$ the set of connected components of $G\backslash S$. A component $C \in \mathcal{C}_G(S)$ is a *full component associated to* S if every vertex of S is adjacent to some vertex of C, i.e. $N_G(C) = S$. The following lemmas (see [15] for a proof) provide different characterizations of a minimal separator:

Lemma 1 *A set S of vertices of G is a minimal a,b-separator if and only if a and b are in different full components of S.*

Lemma 2 *Let G be a graph and S be an a,b-separator of G. Then S is a minimal a,b-separator if and only if for any vertex x of S there is a path from a to b that intersects S only in x.*

If $C \in \mathcal{C}(S)$, we say that $(S,C) = S \cup C$ is a *block* associated to S. A block (S,C) is called *full* if C is a full component associated to S.

Let now $G = (V,E)$ be a graph and $G' = G[V']$ an induced subgraph of G. We will compare the minimal separators of G and G'.

Lemma 3 *Let G be a graph and $V' \subset V$ a vertex set of G. If S is a minimal a,b-separator of the induced subgraph $G' = G[V']$, then there is a minimal a,b-separator T of G such that $T \cap V' = S$.*

Proof. Let $S' = S \cup (V \backslash V')$. Clearly, S' is an a,b-separator in G. Let T be any minimal a,b-separator contained in S'. We have to prove that $S \subseteq T$. Let x be any vertex of S and suppose that $x \notin T$. Since S is a minimal a,b-separator of G', we have a path μ joining a and b in G' that intersects S only in x (see lemma 2). But μ is also a path of G, that avoids T, contradicting the fact that T is an a,b-separator. It follows that $S \subseteq T$. Clearly, $T \cap V' \subseteq S$ by construction of T, so $T \cap V' = S$. $\diamond$

The next corollary follows directly from lemma 3.

Corollary 1 *Let $G = (V,E)$ be a graph and a be a vertex of G. Consider the graph $G' = G[V \backslash \{a\}]$. Then for any minimal separator S' of G', we have that S or $S \cup \{a\}$ is a minimal separator of G. In particular, $|\Delta_G| \geq |\Delta_{G'}|$.*

3 Potential maximal cliques and maximal sets of neighbor separators

The *potential maximal cliques* are the central object of this paper. We present in this section some known results about the potential maximal cliques of a graph (see also [10, 11, 29]).

Definition 1 *A vertex set Ω of a graph G is called a* potential maximal clique *if there is a minimal triangulation H of G such that Ω is a maximal clique of H.*

We denote by Π_G the set of potential maximal cliques of the graph G.

A potential maximal clique Ω is strongly related to the minimal separators contained in Ω. In particular, any minimal separator of G is contained in some potential maximal clique of G. The number $|\Pi_G|$ of potential maximal cliques of G is at least $|\Delta_G|/n$.

If K is a vertex set of G, we denote by $\Delta_G(K)$ the minimal separators of G included in K.

Definition 2 *A set $\mathcal{S}$ of minimal separators of a graph G is called* maximal set of neighbor separators *if there is a potential maximal clique Ω of G such that $\mathcal{S} = \Delta_G(\Omega)$. We also say that* $\mathcal{S}$ borders Ω *in G.*

We proved in [11] that the potential maximal cliques of a graph are sufficient for computing the treewidth and the minimum fill-in of that graph.

Theorem 1 *Given a graph G and its potential maximal cliques Π_G, we can compute the treewidth and the minimum fill-in of G in $\mathcal{O}(n^2|\Delta_G| \times |\Pi_G|)$ time.*

Let now K be a set of vertices of a graph G. We denote by $C_1(K), \ldots, C_p(K)$ the connected components of $G \backslash K$. We denote by $S_i(K)$ the vertices of K adjacent to at least one vertex of $C_i(K)$. When no confusion is possible we will simply speak of C_i and S_i. If $S_i(K) = K$ we say that $C_i(K)$ is a *full component* associated to K. Finally, we denote by $\mathcal{S}_G(K)$ the set of all $S_i(K)$ in the graph G, i.e. $\mathcal{S}_G(K)$ is formed by the neighborhoods, in the graph G, of the connected components of $G \backslash K$.

Consider graph $G = (V, E)$ and a vertex set $X \subseteq V$. We denote by G_X the graph obtained from G by *completing* X, i.e. by adding an edge between every pair of non-adjacent vertices of X. If $\mathcal{X} = \{X_1, \ldots, X_p\}$ is a set of subsets of V, $G_{\mathcal{X}}$ is the graph obtained by completing all the elements of $\mathcal{X}$.

Theorem 2 *Let $K \subseteq V$ be a set of vertices. K is a potential maximal clique if and only if :*

1. *$G \backslash K$ has no full components associated to K.*
2. *$G_{\mathcal{S}_G(K)}[K]$ is a clique.*

Moreover, if K is a potential maximal clique, then $\mathcal{S}_G(K)$ is the maximal set of neighbor separators bordering K, i.e. $\mathcal{S}_G(K) = \Delta_G(K)$.

For example, in figure 1, the vertex sets $\{b, c, e, g\}$ and $\{b, d, e\}$ are potential maximal cliques of the graph of figure 1a and the vertices $\{x, y, z, t\}$ form a potential maximal clique of the graph of figure 1b.

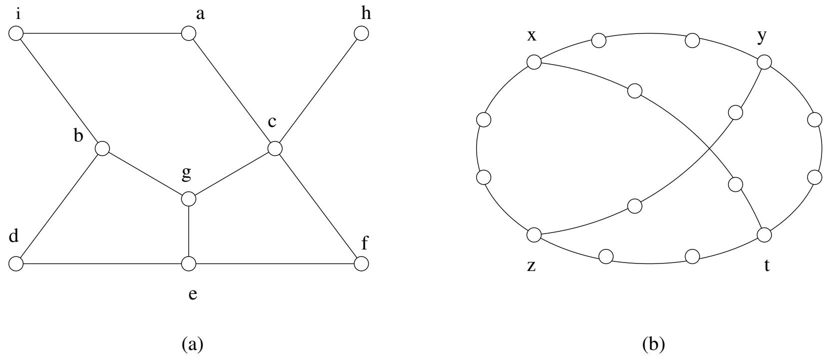

Figure 1: Potential maximal cliques

Remark 1 *If K is a potential maximal clique of G, for any pair of vertices x and y of K either x and y are adjacent in G or they are connected by a path entirely contained in some C_i of $G \backslash K$ except for x and y. The second case comes from the fact that if x and y are not adjacent in G they must belong to the same S_i to ensure that K becomes a clique after the completion of $\mathcal{S}_G(K)$. When we will refer to this property we will say that x* and *y* are connected via the connected component *C_i*.

Remark 2 *Consider a minimal separator S contained in a potential maximal clique Ω. Let us compare the connected components of $G \backslash S$ and the connected components of $G \backslash \Omega$ (see [11] for the proofs). The set $\Omega \backslash S$ is contained in a full component C_Ω associated to S. All the other connected components of $G \backslash S$ are also connected components of $G \backslash \Omega$. Conversely, a connected component C of $G \backslash \Omega$ is either a connected component of $G \backslash S$ (in which case $N_G(C) \subseteq S$) or it is contained in C_Ω (in which case $N_G(C) \not\subseteq S$).*

Remark 3 *Unlike the minimal separators, a potential maximal clique Ω' cannot be strictly included in another potential maximal clique Ω. Indeed, for any proper subset Ω' of a potential maximal clique Ω, the difference $\Omega\backslash\Omega'$ is in a full component associated to Ω'.*

Theorem 2 leads to a polynomial algorithm that, given a vertex set of a graph G, decides if K is a potential maximal clique of G.

Corollary 2 *Given a vertex set K of a graph G, we can recognize in $\mathcal{O}(nm)$ time if K is a potential maximal clique of G.*

Proof. We can compute in linear time the connected components C_i of $G\backslash K$ and their neighborhoods S_i. We can also verify in linear time that $G\backslash K$ has no full components associated to K.

For each $x \in K$, we compute all the vertices $y \in K$ that are adjacent to x in G or connected to x via a C_i in linear time (we have to search the neighborhood of x and the connected components C_i with $x \in S_i$). So we can verify in $\mathcal{O}(nm)$ time if K satisfies the conditions of theorem 2. $\diamond$

4 Potential maximal cliques and active separators

Theorem 2 tells us that if Ω is a potential maximal clique of a graph G, then Ω is a clique in $G_{\Delta_G(\Omega)}$. We will divide the minimal separators of $\Delta_G(\Omega)$ into two classes: those which create edges in $G_{\Delta_G(\Omega)}$, which are called *actives*, and the others, which are called *inactives*. More precisely:

Definition 3 *Let Ω be a potential maximal clique of a graph G and let $S \subset \Omega$ be a minimal separator of G. We say that S is an* active separator for Ω *if Ω is not a clique in the graph $G_{\Delta_G(\Omega)\backslash\{S\}}$, obtained from G by completing all the minimal separators contained in Ω, except S. Otherwise, S is called* inactive for Ω.

Proposition 1 *Let Ω be a potential maximal clique of G and $S \subset \Omega$ a minimal separator, active for Ω. Let (S, C_Ω) be the block associated to S containing Ω and let $x, y \in \Omega$ be two non-adjacent vertices of $G_{\Delta_G(\Omega)\backslash\{S\}}$. Then $\Omega\backslash S$ is an minimal x,y-separator in $G[C_\Omega \cup \{x,y\}]$.*

Proof. Remark that the vertices x and y, non-adjacent in $G_{\Delta_G(\Omega)\backslash\{S\}}$, exist by definition of an active separator. Moreover, since $G_{\Delta_G(\Omega)}$ is a clique, we must have $x, y \in S$.

Let us prove first that $\Omega\backslash S$ is a x,y-separator in the graph $G' = G[C_\Omega \cup \{x,y\}]$. Suppose that x and y are in a same connected component C_{xy} of $G'\backslash(\Omega\backslash S)$. Let $C = C_{xy}\backslash\{x,y\}$. Clearly, $C \subset C_\Omega$ is a connected component of $G\backslash\Omega$. Let T be the neighborhood of C in G. By theorem 2, T is a minimal separator of G, contained in Ω. By construction of T, we have $x, y \in T$. Notice that $T \neq S$, otherwise S would separate C and Ω, contradicting the fact that $C \subset C_\Omega$ (see remark 2). It follows that T is a minimal separator of $\Delta_G(\Omega)$, different from S and containing x and y. This contradicts the fact that x and y are not adjacent in $G_{\Delta_G(\Omega)\backslash\{S\}}$. We can conclude that $\Omega\backslash S$ is an x,y-separator of G'.

We prove now that $\Omega\backslash S$ in a minimal x,y-separator of G'. We will show that, for any vertex $z \in \Omega\backslash S$, there is a path μ joining x and y in G' and such that μ intersects $\Omega\backslash S$ only in z. By theorem 2, x and z are adjacent in $G_{\Delta_G(\Omega)}$, so x and z are adjacent in G or they are connected via a connected component C_i of $G\backslash\Omega$. Notice that $C_i \subset C_\Omega$: indeed, if $C_i \not\subset C_\Omega$, then C_i will be contained in some connected component D of $G\backslash S$, different from C_Ω. According to remark 2, we would have $N_G(C_i) \subseteq N_G(D) \subseteq S$, contradicting $z \in S_i$. In both cases we have a path μ' from x to z in G', that intersects $\Omega\backslash S$ only in z.

For the same reasons, z and y are adjacent in G, or there is a connected component C_j of $G\backslash\Omega$ such that $C_j \subset C_\Omega$ and $z, y \in S_j = N_G(C_j)$. This gives us a path μ'' from z to y in G', such that $\mu'' \cap (\Omega \backslash S) = \{z\}$. Remark that $C_i \neq C_j$, otherwise we would have a path from x to y in $C_i \cup \{x, y\}$, contradicting the fact that $\Omega \backslash S$ separates x and y in G'. So the paths μ' and μ'' are disjoint except for z, and their concatenation is a path μ, joining x and y in G' and intersecting $\Omega \backslash S$ only in z. We conclude by lemma 2 that $\Omega \backslash S$ is a minimal separator of G'. $\diamond$

By proposition 1, the set $T' = \Omega \backslash S$ is a minimal separator of the subgraph of G induced by $C_\Omega \cup \{x, y\}$. By lemma 3, there is a separator T of G such that $T' \subseteq T$ and $T \cap C_\Omega = T'$. We deduce:

Theorem 3 *Let Ω be a potential maximal clique and S be a minimal separator, active for Ω. Let (S, C_Ω) be the block associated to S containing Ω. There is a minimal separator T of G such that $\Omega = S \cup (T \cap C_\Omega)$.*

It follows easily that the number of potential maximal cliques containing at least one active separator is polynomially bounded in the number of minimal separators of G. More exactly number of these potential maximal cliques is bounded by the number of blocks (S, C_Ω) multiplied by the number of minimal separators T, so by $n|\Delta_G|^2$. Clearly, these potential maximal cliques have a simple structure and can be computed directly from the minimal separators of the graph.

Nevertheless, a potential maximal clique may not have active separators. For example in figure 2, the potential maximal clique $\Omega = \{a, c, b', d'\}$ contains the minimal separators $\{a, b', d'\}$, $\{c, b', d'\}$, $\{a, c, b'\}$ and $\{a, c, d'\}$, but no one of them is active for Ω. Let us make a first observation about the potential maximal cliques containing inactive minimal separators.

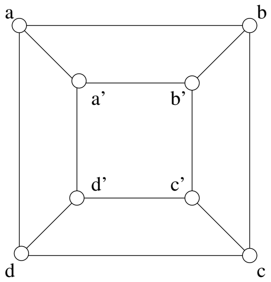

Figure 2: Active and inactive separators

Proposition 2 *Let Ω be a potential maximal clique and $S \subset \Omega$ a minimal separator which is inactive for Ω. Let $D_1, \ldots, D_p$ be the full components associated to S that do not intersect Ω. Then Ω is a potential maximal clique of the graph $G \backslash \cup_{i=1}^{p} D_i$.*

Proof. Let $G' = G \backslash \cup_{i=1}^{p} D_i$. The connected components of $G' \backslash \Omega$ are exactly the connected components of $G \backslash \Omega$, except for $D_1, \ldots, D_p$, and their neighborhoods in G' are the same as in G. It follows that the set $\mathcal{S}_{G'}(\Omega)$ of the neighborhoods of the connected components of $G' \backslash \Omega$ is exactly $\Delta_G(\Omega) \backslash \{S\}$. Clearly, $G' \backslash \Omega$ has no full components associated to Ω. Since S is not active for Ω, we deduce that Ω is a clique in $G'_{\mathcal{S}_{G'}(\Omega)}$. So, by theorem 2, Ω is a potential maximal clique of G'. $\diamond$

5 Removing a vertex

Let $G = (V, E)$ be a graph and a be a vertex of G. We denote by G' the graph obtained from G by removing a, i.e. $G' = G[V \backslash \{a\}]$. We will show here how to obtain the potential maximal cliques of G using the minimal separators of G and G' and the potential maximal cliques of G'. By corollary 1, we know that G has at least as many minimal separators as G': for any minimal separator S of G', either S is a minimal separator of G, or $S \cup \{a\}$ is a minimal separator of G. It will follow that the potential maximal cliques of a graph can be computed in polynomial time in the size of the graph and the number of its minimal separators.

Proposition 3 *Let Ω be a potential maximal clique of G such that $a \in \Omega$. Then $\Omega' = \Omega \backslash \{a\}$ is either a potential maximal clique of G' or a minimal separator of G.*

Proof. Let $C_1, \ldots, C_p$ be the connected components of $G \backslash \Omega$ and S_i be the neighborhood of C_i in G. We denote as usual by $\mathcal{S}_G(\Omega)$ the set of all the S_i's. Remark that the connected components of $G' \backslash (\Omega \backslash \{a\})$ are exactly $C_1, \ldots, C_p$ and their neighborhoods in G' are respectively $S_1 \backslash \{a\}, \ldots, S_p \backslash \{a\}$. Since Ω is a clique in $G_{\mathcal{S}_G(\Omega)}$ (by theorem 2), it follows that $\Omega' = \Omega \backslash \{a\}$ is a clique in $G'_{\mathcal{S}_{G'}(\Omega')}$. If $G' \backslash \Omega'$ has no full components associated to Ω', then Ω' is a potential maximal clique of G', according to theorem 2. Suppose now that C_1 is a full component associated to Ω' in G'. Since C_1 is not a full component associated to Ω in G, it follows that $N_G(C_1) = \Omega'$. Thus, Ω' is a minimal separator of G, by theorem 2. $\diamond$

Lemma 4 *Let G be a graph and $\tilde{G}$ be any induced subgraph of G. Consider a potential maximal clique Ω of $\tilde{G}$. Suppose that for any connected component C of $G \backslash \tilde{G}$, its neighborhood $N_G(C)$ is strictly contained in Ω. Then Ω is also a potential maximal clique of G.*

Proof. Let C be any connected component of $G \backslash \tilde{G}$. We denote by $\tilde{V}$ the set of vertices of $\tilde{G}$. We want to prove that Ω is a potential maximal clique of the graph $\tilde{G}' = G[\tilde{V} \cup C]$. Indeed, the connected components of $\tilde{G}' \backslash \Omega$ are the connected components of $\tilde{G} \backslash \Omega$ plus C. The set $\mathcal{S}_{\tilde{G}'}(\Omega)$ of their neighborhoods consists in $\{N_G(C)\} \cup \mathcal{S}_{\tilde{G}}(\Omega)$. Since $N_G(C)$ is strictly contained in Ω, $\tilde{G}' \backslash \Omega$ has no full components associated to Ω. Obviously Ω is a clique in $\tilde{G}'_{\mathcal{S}_{\tilde{G}'}(\Omega)}$, so Ω is a potential maximal clique of $\tilde{G}'$.

The result follows by an easy induction on the number of connected components of $G \backslash \tilde{G}$. $\diamond$

Proposition 4 *Let Ω be a potential maximal clique of G such that $a \notin \Omega$. Let C_a be the connected component of $G \backslash \Omega$ containing a and let S be the minimal separator of Ω such that $S = N(C_a)$.*

If Ω is not a potential maximal clique of $G' = G[V \backslash \{a\}]$, then S is active for Ω. Moreover, S is not a minimal separator of G'.

Proof. Suppose that S is not active for Ω. Let $D_1, \ldots, D_p$ the full components associated to S in G that do not intersect Ω. One of them, say D_1, is C_a. Let G'' be the graph obtained from G by removing the vertices of $D_1 \cup \ldots \cup D_p$. According to proposition 2, Ω is a potential maximal clique of G''. Notice that G'' is also an induced graph of G'. Any connected component C of $G' \backslash G''$ is contained in some D_i, and its neighborhood in G' is included in $S = N_G(D_i)$. Thus, $N_{G'}(C)$ is strictly contained in Ω. It follows from lemma 4 that Ω is a potential maximal clique of G', contradicting our hypothesis. We deduce that, in the graph G, S is an active separator for Ω.

It remains to show that S is not a minimal separator of G'. We prove that if S is a minimal separator of G', then Ω would be a potential maximal clique of G'. Let $C_1, \ldots . C_p, C_a$ be the connected components of $G \backslash \Omega$ and let $S_1, \ldots, S_p, S$ be their neighborhoods in G. Then the connected components of $G' \backslash \Omega$ are $C_1, \ldots, C_p, C'_1, \ldots, C'_q$, with $C'_i \subset C_a$. Their neighborhoods in G' are respectively $S_1, \ldots, S_p, S'_1, \ldots, S'_q$, with $S'_i \subseteq S$. In particular, $G' \backslash \Omega$ has no full component associated to Ω and $\mathcal{S}_{G'}(\Omega)$ contains every element of $\mathcal{S}_G(\Omega)$, except possibly S. Suppose that S is a minimal separator of G' and let D be a full component associated to S in G', different from C_Ω. By remark 2, D is also a connected component of $G' \backslash \Omega$, so $S = N_{G'}(D)$ is an element of $\mathcal{S}_{G'}(\Omega)$. Therefore, $\mathcal{S}_G(\Omega) \subseteq \mathcal{S}_{G'}(\Omega)$, so Ω is a clique in the graph $G'_{\mathcal{S}_{G'}}(\Omega)$.

We can conclude by theorem 2 that Ω is a potential maximal clique of G', contradicting our choice of Ω. It follows that S is not a minimal separator of G'. $\diamond$

The following theorem, that comes directly from propositions 3 and 4 and theorem 3, shows us how to obtain the potential maximal cliques of G from the potential maximal cliques of G' and the minimal separators of G.

Theorem 4 *Let Ω be a potential maximal clique of G and let $G' = G \backslash \{a\}$. Then one of the following cases holds:*

1. *$\Omega = \Omega' \cup \{a\}$, where Ω' is a potential maximal clique of G'.*
2. *$\Omega = \Omega'$, where Ω' is a potential maximal clique of G'.*
3. *$\Omega = S \cup \{a\}$, where S is a minimal separator of G.*
4. *$\Omega = S \cup (C \cap T)$, where S is a minimal separator of G, C is a connected component of $G \backslash S$ and T is a minimal separator of G. Moreover, S does not contain a and S is not a minimal separator of G'.*

Corollary 3 *Let G be a graph, a be a vertex of G and $G' = G \backslash \{a\}$. The number $|\Pi_G|$ of potential maximal cliques of G is polynomially bounded in the number $|\Pi_{G'}|$ of potential maximal cliques of G', the number $|\Delta_G|$ of minimal separators of G and the size n of G.*

More precisely, $|\Pi_G| \leq |\Pi_{G'}| + n(|\Delta_G| - |\Delta_{G'}|)|\Delta_G| + |\Delta_G|$.

Proof. We will count the potential maximal cliques of the graph G corresponding to each case of theorem 4.

Notice that for a potential maximal clique Ω' of G', only one of Ω' and $\Omega' \cup \{a\}$ can be a potential maximal clique of G: indeed, a potential maximal clique of a graph cannot be strictly included in another one (see remark 3). So the number of potential maximal cliques of type 1 and 2 of G is bounded by $|\Pi_{G'}|$.

The number of potential maximal cliques of type 3 is clearly bounded by $|\Delta_G|$.

Let us count now the number of potential maximal cliques of type 4, that can be written as $S \cup (T \cap C)$. By lemma 3, for any minimal separator S' of G', we have that S' or $S' \cup \{a\}$ is a minimal separator of G. Clearly, the number of minimal separators of G of type S' or $S' \cup \{a\}$ with $S' \in \Delta_{G'}$ is at least $|\Delta_{G'}|$. Our minimal separator S does not contain a and is not a minimal separator of G', so S is not of type S' or $S' \cup \{a\}$, with $S' \in \Delta_{G'}$. It follows that the number of minimal separators S that we can choose is at most $|\Delta_G| - |\Delta_{G'}|$. For each minimal separator S, we have at most n connected components C of $G \backslash S$ and at most $|\Delta_G|$ separators T, so the number of potential maximal cliques of type 4 is at most $n(|\Delta_G| - |\Delta_{G'}|)|\Delta_G|$. $\diamond$

Let now $a_1, a_2, \ldots, a_p$ be an arbitrary ordering of the vertices of G. We denote by G_i the graph $G[\{a_1, \ldots, a_i\}]$, so $G_n = G$ and G_1 has a single vertex. By corollary 3 we have that for any $i, 1 \leq i \leq n$, $|\Pi_{G_{i+1}}| \leq |\Pi_{G_i}| + n(|\Delta_{G_{i+1}}| - |\Delta_{G_i}|)|\Delta_{G_{i+1}}| + |\Delta_{G_{i+1}}|$. Notice that $|\Delta_{G_i}| \leq |\Delta_{G_{i+1}}|$, in particular each graph G_i has at most $|\Delta_G|$ minimal separators. Clearly, the graph G_1 has a unique potential maximal clique. It follows directly that the graph G has at most $n|\Delta_G|^2 + n|\Delta_G| + 1$ potential maximal cliques.

Proposition 5 *The number of the potential maximal cliques of a graph is polynomially bounded in the number of its minimal separators and in the size of the graph.*

More precisely, a graph G has at most $n|\Delta_G|^2 + n|\Delta_G| + 1$ potential maximal cliques.

We give now an algorithm computing the potential maximal cliques of a graph. We suppose that we have a function $IS\_PMC(\Omega, G)$, that returns $TRUE$ if Ω is a potential maximal clique of G, $FALSE$ otherwise.

function ONE_MORE_VERTEX
`Input:` the graphs G, G' and a vertex a such that $G' = G\backslash\{a\}$;
the potential maximal cliques $\Pi_{G'}$ of G', the minimal separators $\Delta_{G'}$, Δ_G of G' and G.
`Output:` the potential maximal cliques Π_G of G.

```
begin
    Π_G ← ∅
    for each p.m.c. Ω' ∈ Π_{G'}
        if IS_PMC(Ω', G) then
            Π_G ← Π_G ∪ {Ω'}
        else
            if IS_PMC(Ω' ∪ {a}, G) then
                Π_G ← Π_G ∪ {Ω' ∪ {a}}
            end_if
        end_if
    end_for
    for each minimal separator S ∈ Δ_G
        if IS_PMC(S ∪ {a}, G) then
            Π_G ← Π_G ∪ {S ∪ {a}}
        end_if
        if (a ∉ S and S ∉ Δ_{G'}) then
            for each T ∈ Δ_G
                for each full component C associated to S in G
                    if IS_PMC(S ∪ (T ∩ C), G) then
                        Π_G ← Π_G ∪ {S ∪ (T ∩ C)}
                    end_if
                end_for
            end_for
        end_if
    end_for
    return Π_G
end
```

Table 1: Computing the p.m.c.'s of G from the p.m.c.'s of $G' = G\backslash\{a\}$

.

The function ONE_MORE_VERTEX of table 1 computes the potential maximal cliques of a graph G from the potential maximal cliques of a graph $G' = G\backslash\{a\}$. This function is based on theorem 4. The main program, presented in table 2, successively computes the potential maximal cliques of the graphs $G_i = G[\{a_1, \dots a_i\}]$. Notice that we can compute the vertex ordering such that each of the graphs G_i is connected.

Theorem 5 *The potential maximal cliques of a graph can be listed in polynomial time in its size and the number of its minimal separators.*

More exactly, the potential maximal cliques of a graph are computable in $\mathcal{O}(n^2m|\Delta_G|^2)$ time.

Proof. Let us analyze the complexity of the algorithm. The sets of vertex sets, like Π_G and Δ_G, will be represented by trees, in such manner that the adjunction of a new element and testing that a vertex set belongs to our set will be done in linear time (see for example [20]). We also know by corollary 2 that a call of the function IS_PMC takes $\mathcal{O}(nm)$ time.

We start with the cost of one execution of the function ONE_MORE_VERTEX.

The cost of the first `for` loop is at most $|\Pi'_G|nm$. But we can strongly reduce this complexity, using a different test for verifying that Ω', respectively $\Omega' \cup \{a\}$ are potential maximal cliques

main program
`Input:` a graph G
`Output:` the potential maximal cliques Π_G of G
`begin`
  let $\{a_1, \ldots, a_n\}$ be the vertices of G
  $\Pi_{G_1} \leftarrow \{\{a_1\}\}$
  $\Delta_{G_1} \leftarrow \emptyset$
  `for` $i = 1, n-1$
    compute $\Delta_{G_{i+1}}$
    $\Pi_{G_{i+1}} = ONE\_MORE\_VERTEX(G_i, G_{i+1}, \Pi_{G_i}, \Delta_{G_i}, \Delta_{G_{i+1}})$
  `end_for`
  $\Pi_G = \Pi_{G_n}$
`end`

Table 2: Algorithm computing the potential maximal cliques

of G. Suppose that we want to check if a potential maximal clique Ω' of G' is also a potential maximal clique of G. Any connected component C' of $G' \backslash \Omega'$ is contained in some connected component C of $G \backslash \Omega$ and we have $N_{G'}(C) \subseteq N_G(C)$. Since Ω' is a clique in the graph $G_{\mathcal{S}_{G'}(\Omega')}$, Ω' is a clique in the graph $G_{\mathcal{S}_G(\Omega')}$. Therefore, all we have to check is that $G \backslash \Omega'$ has no full connected components associated to Ω', which can be done in linear time. Suppose now that Ω' is a potential maximal clique of G' and let us verify if $\Omega = \Omega' \cup \{a\}$ is a potential maximal clique of G. Clearly, the connected components of $G \backslash \Omega$ are the same as the connected components of $G' \backslash \Omega'$. The neighborhood $N_G(C)$ of a connected component of $G \backslash \Omega$ is either $N_{G'}(C)$ or $N_{G'}(C) \cup \{a\}$. It follows that $G \backslash \Omega'$ has no full components associated to Ω and that any two vertices $x, y \in \Omega'$ are adjacent in $G_{\mathcal{S}_G(\Omega)}$. It remains to check that, in the graph $G_{\mathcal{S}_G(\Omega)}$, a is adjacent to any vertex $x \in \Omega'$. This test can be done in linear time: by searching $N_G(a)$ and the connected components C_i of $G \backslash \Omega$ with $a \in S_i$, we compute the vertices of Ω' adjacent to a in G or connected to a via C_i. We conclude that the cost of the first `for` loop is $\mathcal{O}(m|\Pi_{G'}|)$, where $\Pi_{G'} = \mathcal{O}(n|\Delta_G|^2)$.

In the second `for` loop, computing the potential maximal cliques of type 3, i.e. of type $S \cup \{a\}$, costs $\mathcal{O}(nm|\Delta_G|)$ time. This is due to the cost of the Δ_G calls to function IS_PMC. Remark that here we could also test in linear time if $\Omega = S \cup \{a\}$ is a potential maximal clique of G. Since $S \subseteq N_G(C)$ for some connected component of $G \backslash \Omega$ (see proof of proposition 3), we only have to test that $G \backslash \Omega$ has no full components associated to Ω and that a is adjacent in $G_{\mathcal{S}_G(\Omega)}$ to every $x \in S$. Anyway, this will not change the global complexity of the algorithm.

The call to function IS_PMC in the inner loop is done $n|\Delta_G|(|\Delta_G| - |\Delta_{G'}|)$ times. Indeed, we have shown in the proof of corollary 3 that the number of minimal separators $S \in \Delta_G$ such that $a \notin S$ and $S \notin \Delta_{G'}$ is at most $|\Delta_G| - |\Delta_{G'}|$. The number of iteration of the second and third loop are clearly $|\Delta_G|$ and respectively n. So the cost of all the calls to function IS_PMC will be $\mathcal{O}(n^2 m|\Delta_G|(|\Delta_G| - |\Delta_{G'}|)$.

So one execution of the the function ONE_MORE_VERTEX takes at most $\mathcal{O}(nm|\Delta_G|^2 + n^2 m|\Delta_G|(|\Delta_G| - |\Delta_{G'}|)$ time.

We can compute now the complexity of the main program. Computing the minimal separators of a graph G can be done in $\mathcal{O}(n^3|\Delta_G|)$ time, using the algorithm of Berry, Bordat and Cogis [4]. If we do this calculus one time for each graph G_i, this would take $\mathcal{O}(n^4|\Delta_G|)$. But notice that each graph G_i is an induced subgraph of G. Consequently, for each minimal separator S_i of G_i, there is a minimal separator S of G such that $S_i = S \cap \{a_1, \ldots a_1\}$. We can compute first the minimal separators of the input graph G, in $\mathcal{O}(n^3|\Delta_G|)$ time. For computing the minimal separators of a graph G_i, we will take each $S \in \Delta_G$ and we will verify if $S_i = S \cap \{a_1, \ldots a_i\}$ is a minimal separator of G_i. A verification of type $S_i \in \Delta_{G_i}$ can be done in linear time: it is sufficient to test that $G_i \backslash S_i$ has at least two full components associated to S_i (see lemma 1). Therefore, computing the minimal separators of all the graphs G_i will not exceed $\mathcal{O}(n^3|\Delta_G|)$ steps.

Remember that the i-th call of the function ONE_MORE_VERTEX costs at most $\mathcal{O}(nm|\Delta_{G_{i+1}}|^2 + n^2m|\Delta_{G_{i+1}}|(|\Delta_{G_{i+1}}| - |\Delta_{G_i}|)$ time. Using the fact that for all i, $|\Delta_{G_i}| \leq |\Delta_G|$, it follows that the n calls of the function ONE_MORE_VERTEX will take $\mathcal{O}(n^2m|\Delta_G|^2)$ steps.

We conclude that the global complexity of the algorithm is $\mathcal{O}(n^2m|\Delta_G|^2)$. ⋄

We deduce directly from theorem 1, proposition 5 and theorem 5:

Theorem 6 *The treewidth and the minimum fill-in of a graph can be computed in polynomial time in the size of the graph and the number of its minimal separators. The complexity of the algorithm is* $\mathcal{O}(n^3|\Delta_G|^3 + n^2m|\Delta_G|^2)$.

6 Conclusion

The notion of potential maximal clique seems to be very useful for the study of the treewidth and the minimum fill-in problems. We proved in [11] that the potential maximal cliques are sufficient for computing the treewidth and the minimum fill-in of a graph. In this paper, we enumerate the potential maximal cliques in polynomial time in the number of minimum separators of the input graph. In particular, this gives a polynomial algorithm computing the treewidth and the minimum fill-in for all the graphs with polynomial number of minimal separators.

A class of graphs may have an exponential number of minimal separators and consequently an exponential number of potential maximal cliques. Notice that there is no such class of graphs for which the treewidth problem has been solved in polynomial time, except the graphs of bounded treewidth. For example, the problem is still open for the planar graphs. We think that a polynomial number of well-chosen potential maximal cliques could permit to compute or at least approximate the treewidth for classes of graphs with "many" minimal separators.

References

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

[2] S. Arnborg, B. Courcelle, A. Proskurowski, and D. Seese. An algebraic theory of graph reduction. *J. of ACM*, 40:1134–1164, 1993.

[3] S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Applied Mathematics*, 23:11–24, 1989.

[4] A. Berry, J.P. Bordat, and O. Cogis. Generating all the minimal separators of a graph. In *Workshop on Graphs WG'99*, Lecture Notes in Computer Science. Springer-Verlag, 1999.

[5] H. Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11:1–23, 1993.

[6] H. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *Siam J. Computing*, 25:1305–1317, 1996.

[7] H. Bodlaender. Treewidth: Algorithmic techniques and results. In *Proceedings of MFCS'97*, volume 1295 of *Lecture Notes in Computer Science*, pages 19–36. Springer-Verlag, 1997.

[8] H. Bodlaender and B. de Fluiter. Reduction algorithms for constructing solutions of graphs with small treewidth. In *Proceedings of COCOON'96*, volume 1090 of *Lecture Notes in Computer Science*, pages 199–208. Springer-Verlag, 1996.

[9] H. Bodlaender, J.R. Gilbert, H. Hafsteinsson, and T. Kloks. Approximating treewidth, pathwidth, and minimum elimination tree height. *J. of Algorithms*, 18:238–255, 1995.

[10] V. Bouchitté and I. Todinca. Minimal triangulations for graphs with "few" minimal separators. In *Proceedings 6th Annual European Symposium on Algorithms (ESA'98)*, volume 1461 of *Lecture Notes in Computer Science*, pages 344–355. Springer-Verlag, 1998.

[11] V. Bouchitté and I. Todinca. Treewidth and minimum fill-in of weakly triangulated graphs. In *Proceedings 16th Symposium of Theoretical Aspects in Computer Science (STACS'99)*, volume 1563 of *Lecture Notes in Computer Science*, pages 197–206. Springer-Verlag, 1999.

[12] M. S. Chang. Algorithms for maximum matching and minimum fill-in on chordal bipartite graphs. In *ISAAC'96*, volume 1178 of *Lecture Notes in Computer Science*, pages 146–155. Springer-Verlag, 1996.

[13] B. Courcelle. The monadic second-order logic of graphs III: Treewidth, forbidden minors and complexity issues. *Informatique Théorique*, 26:257–286, 1992.

[14] B. Courcelle and M. Moshbah. Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science*, 109:49–82, 1993.

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

[16] T. Hagerup. Dynamic algorithms for graphs of bounded treewidth. In *Proceedings 24th International Colloquium on Automata, Languages, and Programming (ICALP'97)*, Lecture Notes in Computer Science, pages 292–302. Springer-Verlag, 1997.

[17] T. Kloks, H.L. Bodlaender, H. Müller, and D. Kratsch. Computing treewidth and minimum fill-in: all you need are the minimal separators. In *Proceedings First Annual European Symposium on Algorithms (ESA'93)*, volume 726 of *Lecture Notes in Computer Science*, pages 260–271. Springer-Verlag, 1993.

[18] T. Kloks, H.L. Bodlaender, H. Müller, and D. Kratsch. Erratum to the ESA'93 proceedings. In *Proceedings Second Annual European Symposium on Algorithms (ESA'94)*, volume 855 of *Lecture Notes in Computer Science*, page 508. Springer-Verlag, 1994.

[19] T. Kloks and D. Kratsch. Treewidth of chordal bipartite graphs. *J. Algorithms*, 19(2):266–281, 1995.

[20] T. Kloks and D. Kratsch. Listing all minimal separators of a graph. *SIAM J. Comput.*, 27(3):605–613, 1998.

[21] T. Kloks, D. Kratsch, and H. Müller. Approximating the bandwidth for asteroidal triple-free graphs. In *Proceedings Third Annual European Symposium on Algorithms (ESA'95)*, volume 979 of *Lecture Notes in Computer Science*, pages 434–447. Springer-Verlag, 1995.

[22] T. Kloks, D. Kratsch, and J. Spinrad. On treewidth and minimum fill-in of asteroidal triple-free graphs. *Theoretical Computer Science*, 175:309–335, 1997.

[23] T. Kloks, D. Kratsch, and C.K. Wong. Minimum fill-in of circle and circular-arc graphs. *J. Algorithms*, 28(2):272–289, 1998.

[24] A. Parra and P. Scheffler. Characterizations and algorithmic applications of chordal graph embeddings. *Discrete Appl. Math.*, 79(1-3):171–188, 1997.

[25] N. Robertson and P. Seymour. Graphs minors. III. Planar tree-width. *J. of Combinatorial Theory Series B*, 36:49–64, 1984.

[26] N. Robertson and P. Seymour. Graphs minors. II. Algorithmic aspects of tree-width. *J. of Algorithms*, 7:309–322, 1986.

[27] D.J. Rose. Triangulating graphs and the elimination process. *J. Math Anal Appl.*, 32:597–609, 1970.

[28] R. Sundaram, K. Sher Singh, and C. Pandu Rangan. Treewidth of circular-arc graphs. *SIAM J. Discrete Math.*, 7:647–655, 1994.

[29] I. Todinca. *Aspects algorithmiques des triangulations minimales des graphes.* PhD thesis, École Normale Supérieure de Lyon, 1999.

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