A general scheme for deciding the branchwidth

Frédéric Mazoit

Laboratoire de l'Informatique du Parallélisme

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

A general scheme for deciding the branchwidth

Frédéric Mazoit

Juin 2004

Research Report N° RR2004-34

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`

A general scheme for deciding the branchwidth

Frédéric Mazoit

Juin 2004

Abstract

We adapt some decision theorems about treewidth to the branchwidth and use this theorems to prove that the branchwidth of circular-arc graphs can be computed in polynomial time.

Keywords: graphs, branchwidth, circular-arc graphs

Résumé

Nous adaptons des résultats de décision sur les décompositions arborescentes aux décompositions en branches. Nous utilisons ensuite ces résultats pour montrer que le calcul de la largeur de branches des graphes d'intervalles circulaires peut se faire en temps polynomial.

Mots-clés: graphes, largeur de branches, graphes d'intervalles circulaires

1 Introduction

The notion of *treewidth* was introduced by Robertson and Seymour in [14, 15] as a tool for their graph minor theory. This tool has proven to be very fruitful from an algorithmic point of view [2, 3, 4, 9]. A lot of work has been done to compute the treewidth and a corresponding tree decomposition for graphs.

Although the treewidth problem is NP-complete [1], Bouchitté and Todinca proved that it is polynomial when restricted to graphs with a polynomial number of *minimal separators*. To do so they first introduced the notion of *potential maximal clique* which is a clique of a *minimal triangulation* of a graph and proved [5] that the treewidth problem is polynomial when restricted to graphs with a polynomial number of potential maximal cliques. Then they proved [7] that a graph with a polynomial number of minimal separators has a polynomial number of potential maximal cliques and that it is possible to list the potential maximal cliques in polynomial time.

In [16], in an attempt to build an obstruction for the treewidth, Robertson and Seymour defined the *branchwidth*. They proved that for graphs, $\mathrm{bw}(G) \leq \mathrm{tw}(G) + 1 \leq 3/2\, \mathrm{bw}(G)$. Using deep topological results of [17], Seymour and Thomas proved [18] that the branchwidth problem is polynomial for planar graphs which gives the best approximation for the treewidth of planar graphs. In [12], Kloks and al. proved that the branchwidth problem is NP-complete even when restricted to splitgraphs and bipartite graphs. They also gave a polynomial time algorithm to compute the branchwidth of interval graphs.

In this paper we investigate further the links between branchwidth and treewidth by adapting theorems about treewidth to branchwidth. In section 2, we give an overview of the results on treewidth we will extend. We will give slightly modified definitions for some object that do not change for treewidth but will be more convenient later. We will extend the theorems to branchwidth in section 3 and prove that the branchwidth of circular-arc graph is polynomial as an application in 4.

2 Preliminaries

In this paper, we consider simple finite graphs and multigraphs. Let $G = (V, E)$ be a graph or a multigraph. We denote by n the number of vertices of G and m its number of edges. For $V' \subseteq V$, we denote by $N_G(V')$ or $N(V')$ when no confusion is possible the neighbourhood of V' in $G \backslash V'$.

2.1 Treewidth and minimal triangulations

A graph is *chordal* (or *triangulated*) if every cycle of length at least four has a cord, that is an edge between two non-consecutive vertices of the cycle. A *triangulation* of a graph $G = (V, E)$ is a chordal graph $H = (V, E_H)$ such that $E \subseteq E_H$. The triangulation is *minimal* if for every set $E \subseteq E' \subset E_H$, (V, E') is not chordal. A *clique* is a complete subgraph of G.

Property 1 *A chordal graph is the intersection graph of the subtrees of a tree.*

The set of the connected components of $G \backslash S$ is denoted by $\mathcal{C}(S)$. An a, b-*separator* of a graph $G = (V, E)$ is a set $S \subseteq V$ such that a and b are not in the same connected component of $G \backslash S$. We say that S separates a and b. An a, b-*minimal separator* of a graph $G = (V, E)$ is an a, b-separator such that no proper subset of S separates a and b. A minimal separator is a set S which is an a, b-minimal separator for some a and $b \in V$. We denote by Δ_G the set of the minimal separators of G. A connected component C of $G \backslash S$ is a *full component* associated to S if $N(C) = S$. We refer to [11] for the following lemma:

Lemma 1 *A set S of vertices of G is an a, b-minimal separator if and only if a and b are in different full components associated to S. S is a minimal separator if and only if S has at least two full components.*

Let S be a minimal separator of G and $C \in \mathcal{C}(S)$. The set of vertices $(S, C) = S \cup C$ is called a *block*. If S and T are two minimal separators of a graph G, we say that S *crosses* T noted $S\sharp T$ if S separates two vertices $x, y \in T$. If S does not cross T, we say that S and T are *parallel*. The crossing and parallel relation for minimal separators are symmetric and S is parallel to T if S is a subset of a block (T, C). The proof of this statement can be found in [8]. In fact, if we look just a little closer, S crosses T if and only if S intersects $C \in \mathcal{C}(T)$ but S is not included in $C \cup N(C)$. We can thus use "S crosses T if S intersects $C \in \mathcal{C}(T)$ but S is not included in $C \cup N(C)$ as a definition for crossing sets. Note that for general sets, this relation is not symmetric.

Let $X \subseteq V$ we denote by G_X the graph obtained from G by *completing* X, i.e. by adding an edge between every two non adjacent vertices of X. For $\mathcal{X}$ a set of subsets of V, $G^{\mathcal{X}}$ denotes the graph obtained from G by completing the elements of $\mathcal{X}$. The results of [13] establish a strong relation between the minimal triangulation of a graph and its minimal separators.

Theorem 1 *Let $\Gamma \subseteq \Delta_G$ be a maximal set of pairwise parallel minimal separators of G. The graph $H = G^{\Gamma}$ is a minimal triangulation of G and $\Delta_H = \Gamma$.*

Let H be a minimal triangulation of G. The set Δ_H is a maximal set of pairwise parallel minimal separators of G and $H = G^{\Delta_H}$.

A set $\mathcal{S} \subseteq \Delta_G$ is *set of neighbour separators* if for every $S \in \mathcal{S}$, there is $B(S) = (S, C(S))$ such that every $S' \in \mathcal{S}$ is a subset of $B(S)$ and no element of $\mathcal{S}$ contains all the other elements of $\mathcal{S}$. We define the *piece between* $\mathcal{S}$ by $P(\mathcal{S}) = \cap_{S \in \mathcal{S}} B(S)$. If X is a piece between minimal separators, then $\Delta_G(X)$ is the biggest subset of Δ_G such that $X = P(\Delta_G(X))$. We say that the separators in $\Delta_G(X)$ *border* X.

A minimal separator S *splits* a set of neighbour separators $\mathcal{S}$ if $S \subseteq P(\mathcal{S})$, every $S' \in \mathcal{S}$ is a subset of a block (S, C) but no block (S, C) contains all the elements of $\mathcal{S}$. This definition implies that there exists a partition $(\mathcal{S}_1, \ldots, \mathcal{S}_p)$ of $\mathcal{S}$ such that the sets $\mathcal{S}_i \cup \{S\}$ are sets of neighbour separators called the resulting sets. A set $\mathcal{S}$ of neighbour separators is *a maximal set of neighbour separators* if no minimal separator can split $\mathcal{S}$.

Remark 1 *If $\mathcal{S}$ contains only one element or contains an element which contains all the others, there is more than one piece between $\mathcal{S}$ and no minimal separator can split $\mathcal{S}$.*

A *maximal potential clique* of a graph G is a maximal clique of a minimal triangulation of G. The following theorem from [5] gives a characterisation of the maximal potential cliques.

Theorem 2 *A set $\Omega \subseteq V$ is a maximal potential clique if and only if Ω is a piece between a set of maximal neighbour separators and $G^{\Delta_G(\Omega)}[\Omega]$ is a clique.*

Theorem 1 and 2 shows that minimal triangulations are obtained by completing pieces between maximal sets of neighbour separators. If $\mathcal{S}$ is a set of pairwise parallel minimal separators we define $G_{\mathcal{S}}$ by completing in G the pieces between elements of $\mathcal{S}$ that are minimal for inclusion. If the set $\mathcal{S}$ is maximal, then $G_{\mathcal{S}} = G^{\mathcal{S}}$.

If we have a collection $\mathcal{F}$ of maximal potential clique of a graph G, it is natural to ask whether there is a minimal triangulation of G whose maximal cliques all belong to $\mathcal{F}$. In [6], Bouchitté and Todinca defined the notion of *complete family* of maximal potential cliques. That is a family of maximal potential cliques $\mathcal{F}$ such that for any minimal separator included in a clique of $\mathcal{F}$ and any block (S, C), there exists a clique $\Omega \in \mathcal{F}$ such that $S \subset \Omega \subseteq (S, C)$. They proved that given a complete family of maximal potential clique $\mathcal{F}$, one can derive with an *extraction* algorithm, a minimal triangulation H of a graph G whose cliques all belong to $\mathcal{F}$. They also gave an *elimination* algorithm that computes the biggest complete family of maximal potential clique included in a set $\mathcal{F}$. This elimination algorithm 1 has a polynomial running time in n and in $|\mathcal{F}|$.

It turns out that since the family $\mathcal{F}$ only contains maximal potential cliques,

- the set $\Gamma_{\mathcal{F}}$ is the set of minimal separators that border the elements of $\mathcal{F}$;
- a complete family of maximal potential clique is a family of maximal potential clique $\mathcal{F}$ such that for any minimal separator that border a clique of $\mathcal{F}$ and any block (S, C), there exists a clique $\Omega \in \mathcal{F}$ such that $S \subset \Omega \subseteq (S, C)$.

Algorithm 1 Elimination algorithm

Input: A set of maximal potential clique $\mathcal{F}$
The set $\Gamma_{\mathcal{F}}$ of minimal separators of G included if an element of $\mathcal{F}$
Output: The biggest complete family of maximal potential clique included in $\mathcal{F}$
Begin
while there exists $S \in \Gamma_F$ and a block (S, C) such that
no clique $\Omega \in \mathcal{F}$ satisfy $S \subset \Omega \subseteq (S, C)$ **do**
$\mathcal{F} = \mathcal{F} \backslash \{\Omega \in \mathcal{F} \mid S \subset \Omega\}$
update $\Gamma_{\mathcal{F}} = \{S \in \Gamma_{\mathcal{F}} \mid \exists \Omega \in \mathcal{F}$ such that $S \subset \Omega\}$
return $\mathcal{F}$
End

This relaxed definition of complete family can thus be extended to a family of pieces between minimal separators and together with the relaxed definition of $\Gamma_{\mathcal{F}}$, all the proofs used to ensure that the algorithm is correct are still valid. The extraction algorithm also works with the related definitions.

A *tree decomposition* of a graph $G = (V, E)$ is a couple $\mathcal{T} = (T, \chi)$ where T is a tree and χ tags the vertices of T with subsets of V such that:

i. $\forall v \in V$, $\exists w \in V(T)$ with $v \in \chi(w)$;

ii. $\forall (u, v) \in E$, $\exists w \in V(T)$ with $(u, v) \subseteq \chi(w)$;

iii. for any vertex $v \in V$, the vertices $u \in V(T)$ such that $v \in \chi(u)$ induce a subtree T_v of T.

The *width of a tree decomposition* is $\max_{v \in V(T)}\{|\chi(v)| - 1\}$. The *treewidth of a graph* is the minimum width of one of its tree decomposition.

Conditions ii and iii make it natural to see the graph G as a subgraph of the intersection graph $G_{\mathcal{T}}$ of the subtrees T_v. Since the intersection graph of the subtree of a tree is chordal, choosing a tree decomposition of a graph is the same as choosing a triangulation of G and since the width of a tree decomposition corresponds to the size of a maximum clique of $G_{\mathcal{T}}$ minus one, we can suppose that $G_{\mathcal{T}}$ is a minimal triangulation.

Theorem 3 *If a class of graphs $\mathcal{G}$ has a polynomial number of maximal potential cliques and for each graph in $\mathcal{G}$ we can list in polynomial time the set of its maximal potential clique, then the treewidth is polynomial for graphs in $\mathcal{G}$.*

Proof. The algorithm works as follow. First remove from the family $\mathcal{F}$ of the maximal potential clique the elements that have more that $k + 1$ vertices. Then with the elimination algorithm build the biggest complete family in $\mathcal{F}$. If this family is not empty, then the graph has treewidth at most k. Then to build a corresponding tree decomposition, we can use the extraction algorithm on the complete family. □

Remark 2 *In fact we do not need to know all the maximal potential cliques of a graph to be able to compute the treewidth of a graph. We only need a family $\mathcal{F}$ such that if $\mathrm{tw}(G) = k$, then the maximal potential cliques of size at most $k + 1$ in $\mathcal{F}$ contains a non empty complete family.*

2.2 Branchwidth

A *branch decomposition* of a multigraph $G = (V, E)$ is a pair $\Theta = (T, \tau)$ where T is a ternary tree and τ is a bijection from E to the leaves of T. A *branch* of T is a connected component T_i of $T \backslash e$ for $e \in E(T)$. The vertex of degree two of T_i is its root. We extend τ to the branches of T by putting $\tau(T_i) = \{\tau(v) \mid v$ is a leaf of $T_i\}$. The set S_e^{Θ} is the set of vertices x of G such that for each connected component T_i^e of $T\backslash\{e\}$, there exists an edge in $\tau(T_i^e)$ which is incident to x. The *order of an edge* e of T is the size of S_e^Θ. The *width of a branch decomposition* $\Theta = (T, \tau)$ is the maximum order of an edge of T. The *branchwidth* $\mathrm{bw}(G)$ of G is the minimum width of its branch decompositions.

We can see the function τ as a tagging function. This way, we can replace T by a new tree T' that has the same set of leaves and (T', τ) will be a branch decomposition.

The branchwidth and the treewidth are closely related as shown in [16]. Indeed we can transform any tree decomposition without changing the corresponding triangulation in a way that the tree used by the new tree decomposition can be naturally associated to a branch decomposition. We will only prove the first implication.

Lemma 2 *Let $G = (V, E)$ be an hypergraph and $\Theta = (T, \tau)$ be a branch decomposition of G.*

For any three edges e_1, e_2 and e_3 of T such that e_2 is on the path from e_1 to e_3, $S_{e_1}^\Theta \cap S_{e_3}^\Theta \subseteq S_{e_3}^\Theta$.

Proof. Let $T_1^{e_1}$, $T_2^{e_1}$, $T_1^{e_2}$, $T_2^{e_2}$, $T_1^{e_3}$ and $T_2^{e_3}$ be numbered such that

- e_2 and e_3 are edges of $T_2^{e_1}$;
- e_1 is an edge of $T_1^{e_2}$ and e_3 is an edge of $T_2^{e_2}$;
- e_1 and e_2 are edges of $T_1^{e_3}$.

Let $x \in S_{e_1}^\Theta \cap S_{e_3}^\Theta$. There exists $f_1 \in E(T_1^{e_1})$ and $f_2 \in E(T_2^{e_3})$ which are incident to x. By construction, $f_1 \in E(T_1^{e_2})$ and $f_2 \in E(T_2^{e_2})$ which proves that $x \in S_{e_2}^\Theta$. □

Lemma 2 proves that every vertex v of G corresponds to a subtree T_v of T. If for $w \in V(T)$, $\chi(w)$ is the set of vertices v of G such that $w \in V(T_v)$, then (T, χ) is a tree decomposition.

In a way, branchwidth and treewidth are distinct parameters associated to branch decompositions. The branchwidth focuses on the size of the edges of T whereas the treewidth focuses on the vertices of T.

3 Parallel decompositions and branch triangulations

Lemma 3 *Let $\Theta = (T, \tau)$ be a tree decomposition of a multigraph $G = (V, E)$ and e be an edge of T.*

Let C be a connected component of $G\backslash S_e^\Theta$ and $E(C)$ be the set of the edges that are incident to at least one vertex of C.

The set $E(C)$ is a subset of either $E(T_1^e)$ or $E(T_2^e)$.

Proof. Suppose that there exists e_1 and e_2 in $E(C)$ such that $e_i \in E(T_i^e)$. Since C is a connected component of $G\backslash S\Theta_e$, there exists a path $(x_1, \ldots, x_p)$ in C from an end of e_1 to one of e_2. Since e_2 belongs to $E(T_2^e)$, there exists a first x_i which is incident to an edge of $E(T_2^e)$. This vertex belongs to $S\Theta_e$ which is absurd. □

Lemma 3 leads to the definition of a pack. A *pack* of a set of edges $X \subseteq E$ is either an hyperedge whose ends all belong to $\partial(X)$ or a subset $E(C)$ of X for some connected component C of $G\backslash\partial(X)$. A *pack* of $\partial(X)$ is a pack of either X or $E\backslash X$.

For the treewidth, we did not consider tree decompositions that lead to triangulations that were not minimal triangulations. For the branchwidth, we want to do the same thing, i.e. to restrict ourselves to some special branch decompositions that are still optimal. The bond decompositions of [18] are one kind of these decompositions. The decompositions we will consider are *parallel* decompositions. We need to prove that for every multigraph G, there exists such a decomposition whose width is $\mathrm{bw}(G)$. To do so, we will introduce a new parameter associated to branch decompositions which is stronger than the branchwidth and then prove that a decomposition that optimises our parameter is non crossing.

Let $G = (V, E)$ be an hypergraph and $\Theta = (T, \tau)$ be a branch decomposition of G. A *border* of Θ is a set S such that there exists an edge e of T with $S = S_e^\Theta$. A border S is a *primary border* for Θ if it is included in no other border of Θ. The *profile* of Θ is the sequence $(u_n, \ldots, u_1)$ such that u_i is the number of primary borders of Θ of size i. We order the profiles of G with the lexicographical order, that is $(u_n, \ldots, u_1) < (v_n, \ldots, v_1)$ if there exists $i \in [1..n]$ such that $u_i < v_i$ and for every $n \geq j > i$, $u_j = v_j$. The *profile of a graph* G is the minimal profile of a branch decomposition of G. An *optimal branch decomposition* is a branch decomposition whose profile is minimum.

Clearly a branch decomposition whose profile is $\mathrm{profile}(G)$ has width $\mathrm{bw}(G)$.

We want to adapt tools created for the treewidth to the branchwidth. Most of these tools consider minimal triangulation use the fact that the treewidth of a graph is equal to the treewidth of one of its triangulation. The same is true for branchwidth.

Property 2 *Let G be a graph and H a triangulation of G corresponding to a branch decomposition of minimum profile. The graphs G and H have the same profile.*

Proof. First since G is a subgraph of H, any branch decomposition (T, τ) of H can be transformed into a decomposition of G by removing the leaves of T that correspond to edges of $E(H)\backslash E(G)$ and removing the nodes of the resulting tree of degree two. The decomposition has a profile which is not greater that the first one. So $\mathrm{profile}(G) \leq \mathrm{profile}(H)$.

Now consider a branch decomposition $\Theta = (T, \tau)$ of G of minimum profile. The corresponding triangulation H is obtained from G by completing the borders of Θ. We can partition the edges of $E(H)\backslash E(G)$ into subsets $(E_1, \ldots, E_p)$ such that all the edges of E_i belong to the same border S_i. Build a branch with each sets E_i and plug it in the middle of an edge of T corresponding to the border S_i. Since the border of any subset of E_i is included in S_i, the resulting branch decomposition of H and Θ have the same profile. □

Property 2 enables us to try to find a triangulation of G with a minimum branchwidth. Property 5 and theorem 6 give properties of such a triangulation but before that, we need to prove an important theorem.

If one wants to build a branch decomposition Θ with a low profile, it is natural to try to minimise the number of primary borders of Θ and since the packs of $X \subseteq E$ partition X and the border of any union of packs of X is a subset of the border of X, it is natural to try to use them. And indeed properties 7 and 8 prove that this approach works.

Let $\Theta = (T, \tau)$ be a branch decomposition of an hypergraph $G = (V, E)$ and X be a branch of T. The branch X is *split* if there exists a pack B of $\tau(X)$ and no branch Y of X is such that $B = \tau(Y)$.

Theorem 4 *Let $\Theta = (T, \tau)$ be a branch decomposition of an hypergraph $G = (V, E)$ and X be a branch of T. There exists a branch decomposition Θ' such that:*

i. *Θ' is obtained from Θ by replacing X by X' in T;*

ii. *X' is not split;*

iii. $\mathrm{profile}(\Theta') \leq \mathrm{profile}(\Theta)$. *Moreover if there exists a branch W such that $\partial\big(\tau(W)\big)$ crosses $\partial\big(\tau(X)\big)$, then* $\mathrm{profile}(\Theta') < \mathrm{profile}(\Theta)$.

The decomposition Θ' is a decomposition Θ cleaned along X.

The proof of theorem 4 can be found in the appendix.

The following theorem is a direct corollary of theorem 4.

Theorem 5 *Let $G = (V, E)$ be an hypergraph. A branch decomposition Θ of G whose profile is minimum has no two borders that cross.*

A branch decomposition such that no two borders cross is a parallel decomposition.

An important corollary of theorem 5 is the following.

Property 3 *Let $G = (V, E)$ a graph and $\Theta = (T, \tau)$ an optimal branch decomposition. The minimal separators of the triangulation G_Θ are minimal separators of G.*

Proof. Let us prove that if a minimal separator S of G_Θ is not a minimal separator of G then Θ is not optimal.

Since S is a minimal separator of G_Θ, there exists two maximal cliques Ω_1 and Ω_2 of G_Θ such that $S = \Omega_1 \cap \Omega_2$. Moreover maximal cliques correspond to some nodes of T. Let v_1 and v_2 be two vertices of T that correspond to Ω_1 and Ω_2. There exists an edge e of T on the path from v_1 to v_2 that corresponds to S. Let T_1 and T_2 be the two branches of $T \backslash e$ that contain respectively v_1 and v_2. Let $E_i = \tau(T_i)$.

Since S is not a minimal separator of G, there is at most one connected component C whose neighbour is S. We can suppose that no pack B of E_1 is such that $\partial(B) = S$. Let e_1, e_2 and e_3 be the three edges incident to v_1 and S_1, S_2 and S_3 the corresponding borders. In G_Θ, the minimal separator is a strict subset of Ω_1.

Let u be a vertex of $\Omega_1 \backslash S$ and C be the connected component of u in $G \backslash S$. Since $S_3 \subseteq S_1 \cup S_2$, at least one S_i say S_1 contains u but is not included in (S, C) which proves that S_1 crosses S and, by theorem 5 that Θ is not optimal. □

Properties 3 enables us to prove that the maximal cliques of a triangulation corresponding to a parallel branch decomposition are pieces between neighbour separators.

The *branchwidth of a set $\mathcal{S}$ of neighbour separators* is the branchwidth of the clique $P(\mathcal{S})$ to which we add an hyperedge $e_S = \{S\}$ for every minimal separators of $\mathcal{S}$. A *branch clique* of a graph G is the region between a set of neighbour separators.

Theorem 6 *Let $G = (V, E)$ be a graph, $\Theta(T, \tau)$ be parallel branch decomposition of G and G_Θ be the corresponding triangulation of G.*

A maximal clique of G_Θ is a branch clique.

Proof. Let Ω be a maximal clique of G_Θ. Let S_1, S_2 and S_3 be the borders corresponding to the edges incident to a vertex of T associated with Ω. Let $\Delta_{G_\Theta}(\Omega)$ be the set of minimal separators of G_Θ included in Ω. Property 3 proves that $\Delta_{G_\Theta}(\Omega)$ is a set of minimal separators of G.

Let $S \in \Delta_{G\Theta}(\Omega)$. One of S_1, S_2 of S_3 is not included in S. Suppose that S_1 is not included in S. Since S_1 does not cross S, there exists a connected component C such that S_1 is included in $C \cup N(C)$ and so $S_1 \subseteq (S, C)$. But then S_2 and S_3 are also included in (S, C). This implies that $\Omega \subseteq (S, C)$. So the minimal separators of $\mathcal{S} = \Delta_{G_\Theta}(\Omega)$ are indeed neighbour separators and $\Omega \subseteq P_G\big(\mathcal{S}\big)$. In fact, this inclusion is an equality for since G_Θ is a supergraph of G, the region between $\mathcal{S}$ in G is a subset of the one in G_Θ. And in G_Θ, the equality is true. □

Although the problem of finding the branchwidth of a branch clique is exactly the same as finding the branchwidth of a split graph which is NP-complete as shown in [12], if we have a large enough collection of branch cliques whose branchwidth we know, we can use the elimination and the extraction algorithms to decide the branchwidth of a graph. In particular, we have the following theorem:

Theorem 7 *Let $G = (V, E)$ if we can compute in polynomial time a family $\mathcal{F}$ of branch clique such that for every $\Omega \in \mathcal{F}$, we can compute the branchwidth of Ω and such that there exists a triangulation H of G with $\mathrm{bw}(H) = \mathrm{bw}(G)$ and whose maximal clique are all in $\mathcal{F}$, then the branchwidth of G can be computed in polynomial time.*

4 Circular-arc graphs

In this section, we will only consider circular-arc graphs. An *circular arc graph* is the intersection graph of the arcs of a circle. The treewidth of circular-arc graphs can be computed in polynomial time as shown in [19]. To prove this, they use a circular interpretation of the graph (this can be done in polynomial time [10]) and give a geometrical interpretation of maximal potential cliques which allows them to prove that a tree decomposition correspond to a planar triangulation of some polygon. We will follow exactly the same path to prove that the branchwidth of circular-arc graphs can be computed in polynomial time.

We can suppose that the intersection graph $\mathcal{I}$ of a circular-arc graph $G = (V, E)$ is such that the ends of two distinct arcs are also distinct. From now on we will only consider such representations. Between two such ends, be put a *scan point*. A *scan line* is a chord of the circle between two distinct scan points. A *scan triangle* is a triangle between three distinct scan points. It is easy to see that there are $2n$ scan points, $n(2n-1)$ scan lines and $n(2n-1)(2n-2)/3$. The arcs inside of which lie the ends of a scan line are *cut* by the scan line.

Lemma 4 *Let S be a minimal separator of $G = (V, E)$ and $\mathcal{I}$ a representation of G. There exists a scan line of $\mathcal{I}$ that cut exactly S.*

Property 4 *Let Ω be a maximal potential clique of $G = (V, E)$ and $\mathcal{I}$ a representation of G. There exists a scan triangle that cuts exactly Ω.*

Let Θ be a minimal triangulation of G. There exists a plane triangulation Σ of the scan polygon such that the triangles of Σ correspond to all the maximal cliques or subsets of maximal cliques of Θ.

The algorithm of [19] finds a plane triangulation of $\mathcal{I}$ whose biggest triangle cuts a minimal number of arc. And since there are $O(n^3)$ scan triangles, we can use the algorithm of [6] to compute a tree-decomposition.

Consider a branch decomposition $\Theta = (T, \tau)$ of G with a minimum profile, the corresponding triangulation G_Θ and Ω a maximal clique of G_Θ.

Theorems 5 and 6 show that Ω is a branch clique so by property 4, we know that Ω is the piece between the minimal separators that border Ω. Knowing $\Delta_G(\Omega)$ is enough to know Ω. Since $\Delta_G(\Omega)$ is made of minimal separators, by lemma 4, we know that we can represent $\Delta_G(\Omega)$ with a set of scan points. We want to prove that we use at most six scan points to do so. That way, we will have proven that we can enumerate in polynomial time a family of the branch clique that build up the triangulation G_Θ. If we can compute their branchwidth in polynomial time, then by theorem 7 we will have proven that the branchwidth of the circular-arc graph is polynomial. This pattern is exactly the one used in [12]. They prove that an interval graph has a polynomial number of branch clique which they can list. Moreover, they can compute the branchwidth of a branch clique easily.

Property 5 *Let $\Theta = (T, \tau)$ be a branch decomposition of G of minimal profile and Ω a maximal clique of the corresponding triangulation G_Θ. Let e_1, e_2 and e_3 the edges of T incident to a vertex corresponding to Ω.*

There exists at most three scan lines l_j that cut exactly $\Delta_G(\Omega)$ and such that l_j cuts a subset of the border associated to an e_i.

A branch clique of an circular-arc graph that admits such a representation is called a tight branch clique*.*

Proof. Let S_i be the borders associated to e_i.

For $S_j \in \Delta_G(\Omega)$, by lemma 4 let l_j be the scan line between u_j and v_j that cuts exactly S_j. Each S_j is a subset of one S_i. So there exists a set of scan lines $\mathcal{L}$ that cuts exactly $\Delta_G(\Omega)$ and such that each $l_j \in \mathcal{L}$ cuts a subset of one S_i. Moreover, two chords l_j and l_k can only meet at there ends. Choose such a set $\mathcal{L}$ with as few elements as possible.

Let P be the convex polygon defined by the chords of $\mathcal{L}$. Any diagonal of P induce a partition $\mathcal{L}_1 \cup \mathcal{L}_2$ of $\mathcal{L}$. If $|\mathcal{L}| > 3$, then there are two scan lines l_j and l_k in $\mathcal{L}$ that cut the same S_i. This implies that there is a scan line l whose ends lie in the ends of l_j and l_k that is a diagonal of P which induces a partition $\mathcal{L}_1 \cup \mathcal{L}_2$ of $\mathcal{L}$ such that $|\mathcal{L}_1| \geq 2$ and $|\mathcal{L}_2| \geq 2$.

We claim that either $\mathcal{L}_1 \cup \{l\}$ or $\mathcal{L}_2 \cup \{l\}$ is a representation of $\Delta_G(\Omega)$ which is absurd by choice of $\mathcal{L}$. Indeed by theorem 4 we can suppose that the branches T_i are not split. For every scan line s in $\mathcal{L}$, we can build a branch corresponding to connected components of $G\backslash\Omega$ that lie in the cap bordered by s. Now in $\mathcal{L}_1$, take all the scan line s_i that cut a subset of S_1 and create with the corresponding branches a branch corresponding to S_1. Do this for S_2 and S_3 and with the three branches build a branch T_1. Create in a similar way a branch T_2 associated to $\mathcal{L}_2$. By gluing T_1 and T_2 together, we have a new tree decomposition. If neither $\mathcal{L}_1 \cup \{l\}$ nor $\mathcal{L}_2 \cup \{l\}$ represent Ω, then the new branch decomposition has a profile which is strictly smaller than profile(Θ). □

Property 6 *Let Ω be a tight branch clique, l_1, l_2 and l_3 the scan line representing Ω and S_i the set cut by l_i.*

Computing the branchwidth of Ω can be done in polynomial time.

Proof. There is a branch decomposition of width k of the branch clique Ω if and only if there exists a partition $A \cup B \cup C$ of Ω such that $|A \cup B| \leq k$, $|A \cup C| \leq k$ and $|B \cup C| \leq k$, with $S_1 \subseteq A \cup B$, $S_2 \subseteq A \cup B$ and $S_3 \subseteq B \cup C$.

This proves that the branchwidth k of Ω is the minimum integer such that there exists α, β and γ satisfying:

- $|\Omega| = \alpha + \beta + \gamma$;
- $|S_1| \leq \alpha + \beta \leq k$, $|S_2| \leq \alpha + \gamma \leq k$, $|S_3| \leq \beta + \gamma \leq k$;
- $|S_1 \cap S_2| \leq \alpha$, $|S_1 \cap S_3| \leq \beta$ and $|S_2 \cap S_3| \leq \gamma$.

This system can be solved $O(1)$ in $|\Omega|$ and k. □

Property 5 and 6 prove that we can apply theorem 7 and thus the branchwidth problem is polynomial for circular-arc graphs.

5 Conclusion

We have given a framework to compute the branchwidth for classes of graphs. However, this framework is difficult to use. Indeed the number of branch clique of a graph is greater than the number of its maximal potential clique. For instance for a chordal graph G, there are as many branch clique as the number of subtrees of its clique tree. This number is at least exponential in the size of the tree. Knowing this, it is not surprising that the branchwidth of split graphs is NP-complete. We conjecture that the class of chordal graphs whose clique trees have a polynomial number of subtrees, the branchwidth problem is polynomial. Note that interval graph belong to this class.

The work we have conducted seem to show that the branchwidth problem is more difficult that the treewidth problem. The fact that the branchwidth is NP-complete for split graphs whereas the treewidth can be computed in linear time for them confirms this intuition. However, although the branchwidth can be computed in polynomial time for planar graph, the treewidth problem remains open to planar graph. Robertson and Thomas have used deep topological results to solve the branchwidth for planar graphs. We feel that a pure combinatorial approach will not be sufficient to solve the treewidth for planar graphs. Maybe one can find a general algorithm that can solve branchwidth and treewidth for graphs on a surface of fixed genus.

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] H.L. 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.

[5] 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.

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

[7] V. Bouchitté and I. Todinca. Listing all potential maximal cliques of a graph. In *Proceedings 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2000)*, volume 1770 of *Lecture Notes in Computer Science*, pages 503–515. Springer-Verlag, 2000.

[8] Vincent Bouchitté and Ioan Todinca. Approximating the treewidth of at-free graphs. *Discrete Appl. Math.*, 131(1):11–37, 2003.

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

[10] E.M. Eschen and J.P. Spinrad. An $O(n^2)$ algorithm for circular-arc graph recognition. In *Proceedings of SODA'93*, pages 128–137, 1993.

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

[12] T. Kloks, J. Kratochvíl, and H. Müller. New branchwidth territories. In *Proceedings 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS'99)*, volume 1563 of *Lecture Notes in Computer Science*, pages 173–183. Springer-Verlag, 1999.

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

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

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

[16] N. Robertson and P.D. Seymour. Graphs minors. X. Obstruction to tree-decomposition. *J. of Combinatorial Theory Series B*, 52:153–190, 1991.

[17] N. Robertson and P.D. Seymour. Graphs minors. XI. Circuits on a surface. *J. of Combinatorial Theory Series B*, 60:72–106, 1994.

[18] P.D. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.

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

A Appendix

The following property proves that one can suppose that a branch X of a branch decomposition is not split.

Property 7 *Let $\Theta = (T, \tau)$ be a branch decomposition of an hypergraph $G = (V, E)$ and X be a branch of T. There exists a branch decomposition Θ' such that:*

i. *Θ' is obtained from Θ by replacing X by X' in T;*

ii. *X' is not split;*

iii. $\text{profile}(\Theta') \leq \text{profile}(\Theta)$.

The decomposition Θ' is a decomposition Θ cleaned along X.

Proof. Let $X = T^e$. We will prove the lemma by induction on $|X|$,
If $|X| = 1$, then X is not split, we can take $\Theta' = \Theta$.

Otherwise, let Y and Z be the two branches of X.

We may suppose that neither Y nor Z are split. For otherwise, by induction we can change Y in T without increasing the profile and suppose that Y is not split. And then, we can change Z to ensure that Z is not split.

At this point we have a branch decomposition Θ'' in which neither Y nor Z are split and such that $\text{profile}(\Theta'') \leq \text{profile}(\Theta)$. We can suppose that $\Theta = \Theta''$.

Let Y_i (resp. Z_j) be the branches of T that correspond to the packs of $\tau(Y)$ (resp. $\tau(Z)$).
We build X' from the branches Y_i and Z_j in the following way:

- For each pack B of X, let Y_i^B be the branches of Y corresponding to the packs of $\tau(Y)$ included in B. Since $\tau(Y) \subseteq \tau(X)$, a pack of $\tau(Y)$ cannot intersect two packs of $\tau(X)$ so all the leaves of Y appear in the branches Y_i^B. Grow a branch Y^B between the branches Y_i^B. The borders created are the borders of unions of packs of $\tau(Y)$ and thus are included in the border of $\tau(Y)$;
- Create in the same way a branch Z^B;
- Grow a branch T^B between Y^B and Z^B. The created border is the border of B and is included in the border of $\tau(X)$;
- Finally grow the branch X' between the branches T^B and replace T^e by T'^e in T to obtain T'. The borders created are the borders of unions of packs of $\tau(X)$ and thus are included in the border of $\tau(X)$;

We claim that $\Theta' = (T', \tau)$ satisfies the required conditions.
By construction Θ' satisfies condition i. It also satisfies ii for the packs of X' and X are the same and the branches T^B are associated to the packs of X.

As already noted the created borders are subsets of $\partial\big(\tau(Y)\big)$, $\partial\big(\tau(Z)\big)$ or $\partial\big(\tau(X)\big)$ which proves that $\text{profile}(\Theta') \leq \text{profile}(\Theta)$.

□

Property 7 proves that when we clean an edge of a decomposition, we do not increase the profile but in many cases the profile strictly diminishes.

Property 8 *Let $G = (V, E)$ be an hypergraph, $\Theta = (T, \tau)$ be a branch decomposition of G and X a branch of T.*

If there exists a branch W of X such that $\partial\big(\tau(W)\big)$ crosses $\partial\big(\tau(X)\big)$, then Θ cleaned along X has a profile strictly smaller than the profile of Θ.

Proof. Define Y and Z as in lemma 7. We can suppose that $W = Y$ or W is a branch of Y. We prove by induction on the distance between the root of X and W.

If $W = Y$, then by property 7 we can suppose that neither Y nor Z is split. We will prove that any border containing $\partial\big(\tau(Y)\big)$ has been removed from Θ' and that $\text{profile}(\Theta') < \text{profile}(\Theta)$.

Since the packs of $\tau(Y)$ are included in the packs of $\tau(X)$, the borders created to build up the branches Y^B are strictly included in $\partial\big(\tau(Y)\big)$. This proves that no border corresponding to a branch of Y^B contains $\partial\big(\tau(Y)\big)$. We also have that no border corresponding to a branch of Z^B contains $\partial\big(\tau(Y)\big)$. Moreover since $\partial\big(\tau(Y)\big)$ is not included in $\partial\big(\tau(X)\big)$, lemma 2 proves that no border corresponding to a branch of $T\backslash X$ contains $\partial\big(\tau(Y)\big)$. In the end, no branch of Θ' has a border that contains $\partial\big(\tau(Y)\big)$ which proves our claim.

Suppose that $W \neq Y$. Since $\partial\big(\tau(W)\big)$ crosses $\partial\big(\tau(X)\big)$, there exists a connected component C of $G\backslash\partial\big(\tau(X)\big)$ that intersects $\partial\big(\tau(W)\big)$. Since $\tau(W) \subseteq \tau(Y)$, $\partial\big(\tau(Y)\big)$ is not included in (S, C).

- If C is also a connected component of $G\backslash\partial\big(\tau(Y)\big)$, then $\partial\big(\tau(W)\big)$ is not included in $\partial\big(\tau(Y)\big)$. By construction $\partial\big(\tau(W)\big)$ is not included in (S, C). Let C' be another connected component of $G\backslash\partial\big(\tau(Y)\big)$. $\partial\big(\tau(W)\big)$ cannot be included in $C' \cup N(C')$ for $\partial\big(\tau(W)\big)$ intersects C and C avoids $C' \cup N(C')$. This proves that $\partial\big(\tau(W)\big)$ also crosses $\partial\big(\tau(Y)\big)$. By induction hypothesis, the branch decomposition Θ'' defined in property 7 is such that $\text{profile}(\Theta') \leq \text{profile}(\Theta'') < \text{profile}(\Theta)$.

- If C is not a connected component of $G\backslash\partial\big(\tau(Y)\big)$, then $\partial\big(\tau(Y)\big)$ intersects C. So for any other connected component C' of $G\backslash\partial\big(\tau(X)\big)$, since $C \cap \big((S, C)\big) = \emptyset$, $\partial\big(\tau(Y)\big)$ is not included is (S, C). This proves that $\partial\big(\tau(Y)\big)$ crosses $\partial\big(\tau(X)\big)$.

$\square$