N° d'ordre : 105
N° attribué par la bibliothèque : 99ENSL0105

# THÈSE

présentée par

Ioan Todinca

devant
l'ÉCOLE NORMALE SUPÉRIEURE DE LYON

pour l'obtention du
DIPLÔME DE DOCTORAT
Spécialité : Informatique

# Aspects algorithmiques des triangulations minimales des graphes

Date de soutenance : 15 janvier 1999

Composition du Jury :

| | |
|---|---|
| Vincent Bouchitté, | Directeur de thèse |
| Bruno Courcelle, | Rapporteur |
| Michel Habib, | Rapporteur |
| Jacques Mazoyer | |
| Henri Thuillier | |
| Brigitte Vallée | |

Thèse préparée à l'École Normale Supérieure de Lyon,
au sein du Laboratoire de l'Informatique du Parallélisme,
Unité Mixte de Recherche CNRS-INRIA-ENS LYON n° 8512.

*A Alice,*
*à mon frère,*

## Remerciements

Il paraît que j'ai rencontré pour la première fois Vincent Bouchitté en 94, plus exactement il m'a fait passer un oral. Lorsque je l'ai revu, deux ans plus tard, pour lui demander un stage de DEA, je ne l'ai pas reconnu – lui, si. Qu'il me pardonne pour cela – à l'époque je n'étais pas un étudiant très assidu (j'en avais même une dispense)...

Maintenant que je le connais un peu mieux, je voudrais d'abord le remercier pour la confiance qu'il m'a accordée depuis trois ans déjà. Je lui suis également reconnaissant pour toutes les discussions, techniques ou pas, que nous avons eues pendant cette période, et pour l'extrême patience dont il a fait preuve tout au long la rédaction de ce document.

Je suis redevable aux deux rapporteurs, Bruno Courcelle et Michel Habib, pour les remarques autour de ma thèse. J'ai été très touché par l'intérêt qu'ils ont porté à mon travail et par leurs commentaires.

Je remercie Brigitte Vallée et Henri Thuillier d'avoir accepté d'être membres de mon jury.

Que Jacques Mazoyer sache que je lui suis particulièrement reconnaissant d'avoir été mon directeur administratif et d'être membre du jury, mais surtout pour la bonne humeur qu'il sait transmettre à ceux qui le côtoient.

Ces années n'auraient pas eu la même saveur sans la compagnie de mes compères thésards. C'est aussi grâce à eux que l'on peut alterner le travail dans une excellente ambiance avec des moments de détente tout aussi plaisants. Je commence avec les anciens, Ivan Rapaport et Enrico Formenti, avec lesquels j'ai passé d'agréables (et longues, je l'avoue) pauses café chez "Esther". Et merci à mes chers co-bureaux, Codrin Nichitiu et Grégory Lafitte, pour leur joie quasi-permanente.

Je remercie Marianne Delorme et Éric Rémila pour les nombreuses discussions et leurs conseils, et Bruno Durand pour me supporter, pour la deuxième année consécutive, comme TD-man.

Je tiens à remercier le LIP pour son ambiance stimulante et pour l'efficacité de son organisation. Je suis particulièrement reconnaissant envers Nicole Nourry pour sa gentillesse et son aide pour tous les problèmes administratifs.

Je dois beaucoup à mes parents et à mon frère pour leur soutien moral à distance, pour leurs messages électroniques restés souvent sans réponse pendant les périodes chargées.

A Alice, je présente mes excuses pour mes humeurs parfois peu propices à la détente, en particulier durant la rédaction. Je te remercie pour ton appui et je te promets de me rattraper de mon mieux...

# Table des matières


**1 Introduction** **3**
1.1 Calcul de la largeur arborescente et d'autres paramètres : état de l'art 6
1.2 Résumé de la thèse . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

**2 Préliminaires** **11**
2.1 Terminologie . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
2.1.1 Graphes quelconques . . . . . . . . . . . . . . . . . . . . . . . 11
2.1.2 Classes de graphes . . . . . . . . . . . . . . . . . . . . . . . . 12
2.2 Graphes triangulés . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
2.3 Problèmes de triangulation . . . . . . . . . . . . . . . . . . . . . . . 15
2.4 Séparateurs minimaux . . . . . . . . . . . . . . . . . . . . . . . . . . 17
2.4.1 Généralités sur les séparateurs minimaux . . . . . . . . . . . 17
2.4.2 Séparateurs minimaux des graphes triangulés . . . . . . . . . 19
2.4.3 Énumération des séparateurs minimaux d'un graphe . . . . . 21
2.5 Triangulations minimales et séparateurs . . . . . . . . . . . . . . . . 23

**3 Triangulations minimales** **31**
3.1 Cliques maximales des graphes triangulés . . . . . . . . . . . . . . . 32
3.2 Cliques maximales potentielles . . . . . . . . . . . . . . . . . . . . . 35
3.3 Une autre caractérisation des cliques maximales potentielles . . . . . 40
3.4 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

**4 Largeur arborescente et complétion minimale** **47**
4.1 Calcul de la largeur arborescente . . . . . . . . . . . . . . . . . . . . 48
4.1.1 Description de l'algorithme . . . . . . . . . . . . . . . . . . . 48
4.1.2 Implémentation et complexité . . . . . . . . . . . . . . . . . . 54
4.2 Algorithmes de programmation dynamique . . . . . . . . . . . . . . . 57
4.2.1 Triangulations des blocs . . . . . . . . . . . . . . . . . . . . . 57
4.2.2 Implémentation et complexité . . . . . . . . . . . . . . . . . . 61
4.3 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62

**5 Application à des classes de graphes** **65**
5.1 Graphes sans triplet astéroïde et leur généralisation . . . . . . . . . . 66
5.2 Graphes de cordes et graphes d'intervalles circulaires . . . . . . . . . 70

5.3 Graphes faiblement triangulés . . . . . . . . . . . . . . . . . . . . . . . . 78
5.4 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85

**6 Conclusion et perspectives** **87**

**Bibliographie** **89**

**Index** **94**

# Chapitre 1

# Introduction

La notion de *largeur arborescente* – en anglais : *treewidth* – a été introduite par Robertson et Seymour au début des années 80 [RS86, RS84]. Elle est fortement liée au concept de $k$-arbre partiel, que l'on trouve déjà dans les années 70 dans les travaux de Dirac et Rose [Ros74]. Depuis lors, une quantité impressionnante de travaux sont liés ou s'appuient sur la largeur arborescente.

La volumineuse théorie qui a conduit à cette notion est fort bien illustrée par le plongement des graphes non orientés sur certains types de surfaces. Prenons comme exemple le cas des graphes planaires, c'est-à-dire les graphes que l'on peut le dessiner dans le plan sans que deux arêtes se croisent. Considérons trois opérations sur les graphes : la première consiste à enlever une arête du graphe, la deuxième, à lui enlever un sommet et les arêtes adjacentes, et enfin la troisième est de contracter une arête en un seul sommet. Si l'on applique ces opérations à un graphe planaire, il est assez facile de se convaincre que nous obtenons toujours un graphe planaire. Plus formellement, un graphe $H$ est appelé *mineur* de $G$ si $H$ est obtenu à partir de $G$ par une suite d'opérations de ce type. Comme nous l'avons remarqué, la minoration préserve la planarité. Nous dirons que les graphes planaires constituent une classe de graphes fermée pour la minoration.

Les plus petits graphes non planaires sont le graphe complet à cinq sommets, noté $K_5$, et le graphe biparti complet $K_{3,3}$ de la figure 1.1. Un théorème fort connu dû à Kuratowski nous assure que tout graphe $G$ non planaire a $K_5$ ou $K_{3,3}$ comme mineur. D'un autre côté, un graphe planaire ne peut pas avoir comme mineur un de ces deux graphes. En d'autres termes, les graphes planaires peuvent être identifiés comme la classe des graphes n'ayant pas de mineur dans l'ensemble $\{K_5, K_{3,3}\}$. Les deux mineurs interdits forment ce que l'on nomme *ensemble d'obstructions* pour les graphes planaires.

De la même manière que pour les graphes planaires, on peut définir les graphes représentables sur d'autres surfaces, par exemple les tores. Toute classe définie de cette manière à partir d'une surface est fermée par minoration.

Wagner pose déjà en 1937 le problème de savoir si une classe de graphes fermée par minoration peut être caractérisée par un ensemble fini d'obstructions, comme

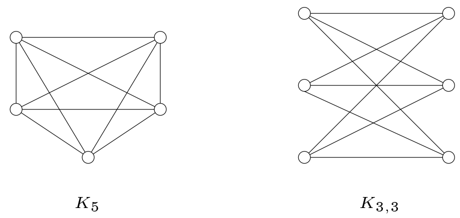


FIG. 1.1: Ensemble d'obstructions des graphes planaires

dans le cas des graphes planaires et du théorème de Kuratowski. Étant donnée une classe de graphes $\mathcal{G}$ fermée par minoration, existe-t-il un ensemble fini de graphes $ob(\mathcal{G})$ telle qu'un graphe $G$ appartient à $\mathcal{G}$ si et seulement si aucun graphe de $ob(\mathcal{G})$ n'est mineur de $G$ ? Cette conjecture a été prouvée par Robertson et Seymour et c'est l'un des résultats principaux de la longue série d'articles qu'ils ont consacrée à l'étude des mineurs de graphes.

C'est dans ce cadre-là qu'est apparue la notion de *largeur arborescente*, dans le deuxième article de la série. La largeur arborescente est une mesure de connexité : si un graphe a une largeur arborescente petite cela indique que le graphe est "peu connexe", qu'il a une structure proche d'un arbre, ou plutôt d'une notion d'arbre généralisé que nous détaillerons plus tard. La largeur arborescente a été introduite afin de caractériser certaines classes de graphes fermées par minoration, ainsi que pour les reconnaître de façon algorithmique. Un des premiers résultats importants est que si l'on se donne un graphe planaire $H$, tout graphe $G$ n'ayant pas $H$ comme mineur a une largeur arborescente bornée par une constante ne dépendant que de $H$ [RS84]. Les graphes de largeur arborescente au plus $k$ sont eux-mêmes une classe fermée par minoration.

Il est à noter que, pour la classe de graphes de largeur arborescente bornée par une constante, on peut vérifier en temps linéaire si un élément de la classe a un certain graphe $H$ comme mineur. Il existe depuis relativement peu de temps un algorithme qui décide en temps linéaire si la largeur arborescente d'un graphe est plus petite qu'une certaine constante [Bod96]. On pourrait en conclure que, étant donnée une classe de graphes $\mathcal{G}$, fermée par minoration et qui ne contient pas tous les graphes planaires, alors pour tout graphe $G$ nous pouvons décider en temps linéaire si $G$ appartient à cette classe.

Tous les résultats algorithmiques sur les mineurs des graphes ont tout de même l'important défaut d'être non constructifs. En effet, lorsque l'on se donne une classe de graphes fermée par minoration, pour appliquer ces algorithmes il faudrait d'abord trouver son ensemble d'obstructions. Même pour des classes naturelles et relativement simples, trouver l'ensemble d'obstructions est un problème ouvert. De plus, les constantes que cachent les algorithmes sont presque toujours très grandes.

Mais la largeur arborescente a concentré l'intérêt des algorithmiciens dans un autre contexte. Comme nous l'avons dit, si un graphe a une largeur arborescente petite, c'est qu'il a une structure proche d'un arbre. Il existe alors une *décomposition arborescente* "peu large" du graphe, c'est-à-dire une représentation du graphe comme une sorte d'arbre généralisé.

Il a été constaté assez tôt que certains problèmes réputés difficiles sont résolubles pour des graphes de largeur arborescente petite. Beaucoup de problèmes classiques de l'algorithmique des graphes, qui sont NP-difficiles en général, peuvent être résolus en temps polynomial, voire linéaire, pour les graphes ayant une largeur arborescente bornée par une constante. On y retrouve par exemple le problème du stable maximum ou du cycle hamiltonien. Il existe des résultats qui identifient des classes de problèmes résolubles en temps linéaire pour des graphes de largeur arborescente bornée. Les travaux de Courcelle [Cou92], Arnborg et al. [AP89], Courcelle et Mosbah [CM93] ont abouti à la conclusion que tous les problèmes exprimables par des formules monadiques étendues de second ordre peuvent être résolus en temps linéaire. Schématiquement, les problèmes de graphes que l'on peut formuler à l'aide des opérateurs logiques ($\wedge$, $\vee$, $\neg$, $\Rightarrow$), des quantificateurs sur des sommets, arêtes, ensembles de sommets et ensemble d'arêtes (comme $\exists W \subseteq V$, $\forall e \in E$), de tests d'appartenance et d'adjacence ($v \in W$, $xy \in E$) et certaines extensions peuvent être résolus pour des graphes de largeur arborescente bornée. Par exemple, le problème de décider si un graphe $(V, E)$ est trois-coloriable peut être exprimé de cette façon :

$\exists W_1 \subseteq V\ : \exists W_2 \subseteq V\ : \exists W_3 \subseteq V\ : (\forall v \in V\ : v \in W_1 \vee v \in W_2 \vee v \in W_3) \wedge \forall v \in V\ : \forall w \in V\ : vw \in E \Rightarrow (\neg(v \in W_1 \wedge w \in W_1) \wedge \neg(v \in W_2 \wedge w \in W_2) \wedge \neg(v \in W_3 \wedge w \in W_3))$.

Les extensions permettent de prendre en considération des problèmes d'optimisation, comme le problème du stable maximum.

Pour résoudre des problèmes sur des graphes de largeur arborescente bornée on procède souvent par programmation dynamique [Bod93, Hag97]. Dans ce cas, l'algorithme a besoin de connaître la largeur arborescente du graphe, ainsi qu'une décomposition arborescente. Une décomposition arborescente d'un graphe de largeur arborescente $k$ est un arbre étiqueté par des ensembles de sommets du graphe, de sorte que chaque étiquette ait au plus $k + 1$ sommets. L'idée de ce type d'algorithme est de parcourir l'arbre de décomposition des feuilles vers l'intérieur et de calculer au fur et à mesure des solutions partielles pour le sous-graphe induit par les étiquettes rencontrées.

Une autre technique est d'employer des algorithmes de réduction [ACPS93, BdF96], qui fonctionnent sans connaître les décompositions arborescentes. Pour vérifier si un graphe a une certaine propriété, on procède en le transformant, par une série d'opérations de réductions qui préservent ladite propriété, en un graphe plus petit. On réduit ainsi le graphe jusqu'à ce qu'aucune règle de réduction ne soit plus appliquable. Il suffit alors de tester si le graphe ainsi obtenu fait partie d'une famille finie de graphes, auquel cas le graphe en entrée avait la propriété cherchée. Si le résultat des réductions successives ne se trouve pas dans cet ensemble fini, on décide que le graphe en entrée ne satisfait pas la propriété. Cette technique s'applique à des problèmes de décision, et elle peut même être adaptée à des problèmes d'optimisation.

Une présentation générale de ce type de techniques est donnée par Bodlaender dans [Bod93, Bod97].

Nous avons dit que ces algorithmes sont linéaires pour des graphes de largeur arborescente bornée. Pour être plus précis, pour résoudre un problème comme le stable maximum il faut un temps de $\mathcal{O}(2^k n)$, lorsque le graphe en entrée a $n$ sommets et la largeur arborescente au plus $k$. Ceci donne une idée de la borne supérieure que l'on peut accepter pour $k$ de sorte à rester à des complexités "raisonnables".

La notion de largeur arborescente répond au besoin de donner une structure simple, facile à maîtriser, à des objets complexes. Si nous pouvons ramener des graphes d'apparence complexe à être vus comme une sorte d'arbres, nous transformons de la même manière certains problèmes sur ces objets, de "difficiles" (NP-difficiles) à "très simples" (linéaires).

Le cadre de cette thèse est la recherche de structure de ce type dans de larges classes de graphes, de trouver de "bonnes" décompositions arborescentes pour ces classes. Nous chercherons donc à calculer la largeur arborescente - et d'autres paramètres liés à la largeur arborescente - pour certains types de graphes.

## 1.1 Calcul de la largeur arborescente et d'autres paramètres : état de l'art

Pour résoudre des problèmes sur des graphes de largeur arborescente "petite", nous avons souvent besoin de connaître la largeur arborescente et une décomposition arborescente du graphe en entrée. Mais calculer la largeur arborescente d'un graphe est un problème NP-difficile [ACP87], même pour des classes restreintes comme les graphes de degré borné [BT95], les graphes bipartis ou les graphes de cocomparabilité [HM94].

Comme nous l'avons mentionné, il existe un algorithme qui, prenant en entrée un graphe $G$ à $n$ sommets et une constante $k$, décide en temps $\mathcal{O}(n)$ si la largeur arborescente de ce graphe est au plus $k$. Cet algorithme est dû à Bodlaender [Bod96]. Pour nos applications NP-difficiles en général, mais de complexité $\mathcal{O}(n)$ pour la classe des graphes de largeur arborescente au plus $k$, la constante cachée par le $\mathcal{O}$ est de l'ordre de $2^k$. On pourrait en conclure que, puisque ce qui nous intéresse est de savoir si un graphe a une largeur arborescente "petite", l'algorithme de Bodlaender est suffisant. Mais le facteur que cache la notation $\mathcal{O}$ pour cet algorithme est sur-exponentiel en $k$, la complexité de l'algorithme de Bodlaender étant en fait de $\mathcal{O}(2^{(2k)^2}n)$. Par conséquent, ce procédé ne peut pas être employé dans la pratique, même pour des constantes $k$ très petites.

Il existe des algorithmes beaucoup plus performants pour décider si la largeur arborescente d'un graphe est au plus 2, 3 ou 4, basés directement sur le fait qu'une classe de graphes de largeur arborescente bornée est fermée par minoration et qui fonctionnent par des techniques de réduction [AP86, San96]. Ces algorithmes n'ont pas été étendus à ce jour, ne serait-ce que pour la classe des graphes de largeur arborescente au plus 5.

Néanmoins, pour bon nombre de classes de graphes particulières, il existe des algorithmes polynomiaux qui calculent la largeur arborescente des graphes respectifs. Les techniques employées sont complètement différentes de celles mentionnées plus haut. Avant de parler de ces algorithmes, revenons sur la notion de largeur arborescente.

Nous avons dit qu'un graphe de largeur arborescente petite est un graphe proche d'un "arbre généralisé". Les arbres généralisés sont en fait les graphes triangulés, c'est-à-dire les graphes tels que tout cycle d'au moins quatre sommets possède une corde. Mais ce qui nous intéresse ici est le fait que les graphes triangulés peuvent être représentés comme des arbres étiquetés, où les étiquettes sont les cliques maximales du graphe. Il est naturel de définir la largeur arborescente d'un tel graphe comme étant la taille de l'étiquette la plus grande moins un. Le "moins un" est fait pour que les arbres aient une largeur arborescente égale à 1. Si l'on note $\omega(G)$ la taille de la clique de cardinal maximum de $G$, pour un graphe triangulé $H$ on a une largeur arborescente, notée $\mathrm{tw}(H)$, égale à $\omega(H)-1$. Si l'on considère maintenant un graphe quelconque, on peut lui donner une *décomposition arborescente* en le plongeant dans un graphe triangulé. La largeur arborescente d'un graphe quelconque est définie comme la plus petite largeur arborescente de toutes les décompositions arborescentes du graphe.

Le problème du calcul de la largeur arborescente se ramène à la question suivante : étant donné un graphe $G$, comment trouver un graphe $H$ contentant $G$, de sorte que $H$ soit triangulé et que $\omega(H)$ soit minimum ? Nous cherchons donc des graphes triangulés contenant $G$, que l'on appelle aussi des triangulations de $G$. Nous pouvons nous restreindre à des triangulations $H$ minimales par inclusion, en bref à des triangulations minimales.

Il existe un autre problème très naturel lié aux triangulations minimales. C'est le problème de la *complétion minimale* d'un graphe : étant donné un graphe quelconque $G$, trouver une triangulation $H$ de $G$ de sorte que le nombre d'arêtes de $H$ soit minimum. Le problème de la complétion minimale est intéressant dans le cadre de la factorisation des matrices creuses [Ros70] et il est, comme la largeur arborescente, NP-difficile pour les graphes quelconques [Yan81].

Pour résoudre chacun de ces problèmes, nous cherchons des triangulations d'un graphe, tout en minimisant une certaine quantité. Il est naturel que les techniques pour résoudre ces questions soient très proches. Le tableau 1.1 contient des classes de graphes pour lesquelles les deux problèmes ont été résolus. Nous avons indiqué les références pour chacun des deux problèmes - largeur arborescente et complétion minimale - et nous pouvons observer que pour les travaux récents les deux questions sont traitées simultanément. Pour certaines de ces classes, les mêmes techniques conduisent aussi au calcul d'autres paramètres de triangulation comme la largeur arborescente linéaire (pathwidth) ou le vertex ranking, que nous ne traiterons pas dans cette thèse.

Tous les algorithmes cités calculent des triangulations minimales des graphes,

| *classe de graphes* | *largeur arborescente et complétion minimale* |
|---|---|
| graphes triangulés | direct |
| graphes de permutation | [BKK95, KKS97] |
| graphes AT-free avec nombre polynomial de séparateurs | [KKS97, KKS97] |
| graphes d'intervalles circulaires | [SSR94, KKW98] |
| graphes bipartis cordaux | [KK95, Cha96] |
| graphes de cordes | [Klo96, KKW98] |
| graphes distance héréditaires | [BDK97b, BDK97b] |
| graphes HHD-free | [BDK97a, BDK97a] |

TAB. 1.1: Classes de graphes pour lesquelles le problème de la largeur arborescente et de la complétion minimale est polynomial

même si les tout premiers sont rédigés en employant des notions quelque peu différentes. La résolution de ces problèmes passe par la compréhension des triangulations minimales des graphes quelconques. Plus intéressant, tous ces graphes ont un point commun : ils ont un nombre polynomial de séparateurs minimaux.

Une tentative pour donner un algorithme qui calcule la largeur arborescente et la complétion minimale de tous les graphes ayant un nombre polynomial de séparateurs a été faite dans [KBMK93], mais malheureusement l'algorithme est faux [KBMK94]. Néanmoins, le problème de savoir si l'on peut calculer en temps polynomial la largeur arborescente et la complétion minimale des graphes avec "peu" de séparateurs est un des principaux problèmes ouverts du domaine. Nous nous référerons à cette question comme "la conjecture ESA'93".

Beaucoup de travaux sont dédiés à cette conjecture. Le but est de comprendre la relation entre les séparateurs minimaux et les triangulations minimales d'un graphe. Les premières intuitions apparaissent déjà dans les travaux de Bodlaender, Kloks et Kratsch [BKK95] qui travaillent sur des classes particulières. Une première caractérisation puissante des triangulations minimales en termes de séparateurs minimaux est donnée par Kloks, Kratsch et Müller [KKM95], et elle est complétée par Parra et Scheffler dans [PS97]. Dans tous ces articles sont données des caractérisations *globales* des triangulations minimales des graphes, en regardant *globalement* la structure des séparateurs minimaux. C'est pourquoi ces caractérisations n'ont pas de conséquence algorithmique directe sur le calcul de la largeur arborescente et de la complétion minimale des graphes.

Nous avons voulu apporter un point de vue *local*, donc plus algorithmique, sur les triangulations minimales des graphes. Nous avons cherché des objets qui caractérisent complètement les triangulations minimales d'un graphe, et qui soient en même temps calculables au moins pour les graphes cités.

Les notions que nous avons introduites, de *famille maximale de séparateurs voisins* et celle de *clique maximale potentielle* caractérisent les ensembles de sommets d'un graphe $G$ qui forment des cliques maximales dans les triangulations minimales de

$G$ (signalons que la terminologie de "clique maximale potentielle" est reprise d'après Kloks, Kratsch et Müller [KKM98], qui ont travaillé sur ces questions indépendamment). Nous montrerons comment reconnaître ces objets localement, sans passer par les triangulations minimales mais en restant dans le graphe $G$. Nous prouverons que, une fois que les cliques maximales potentielles d'un graphe sont énumérées, la largeur arborescente et la complétion minimale du graphe peuvent être calculées en un temps qui dépend polynomialement de la taille de $G$ et du nombre de ses cliques maximales potentielles. Enfin, nous verrons que pour toutes les classes de graphes du tableau 1.1, nous pouvons énumérer les cliques potentielles en temps polynomial, et donc calculer la largeur arborescente et la complétion minimale en temps polyonmial, ce qui unifie les algorithmes mentionnés. De plus, nous donnerons un algorithme calculant ces deux paramètres pour une nouvelle classe de graphes, à savoir les graphes faiblement triangulés.

## 1.2 Résumé de la thèse

Le document est organisée en 6 chapitres.

Le chapitre 2 introduit formellement les définitions de base de cette thèse et passe en revue les résultats connus sur les triangulations minimales. Nous parlerons de représentations arborescentes des graphes triangulés et nous ferons la relation entre triangulations minimales, largeur arborescente et complétion minimale. Nous regarderons ensuite la notion de séparateur minimal, en présentant en détail la liaison entre les représentations arborescentes et les séparateurs minimaux d'un graphe triangulé. Nous montrerons que l'on peut énumérer tous les séparateurs minimaux d'un graphe, en un temps qui dépend de façon polynomiale de la taille du graphe et du nombre de ses séparateurs minimaux. Enfin, nous donnerons une caractérisation globale des triangulations minimales d'un graphe à l'aide de ses séparateurs minimaux. Ce chapitre est uniquement constitué de résultats déjà connus.

Le chapitre 3 est au cœur de cette thèse. Nous introduisons deux notions, les *familles maximales de séparateurs voisins* et les *cliques maximales potentielles*, qui permettent de caractériser complètement les triangulations minimales d'un graphe. Les cliques maximales potentielles sont exactement les ensembles de sommets d'un graphe $G$ qui forment une clique maximale dans les triangulations minimales de $G$. Nous verrons que, en dehors de leur définition de nature combinatoire, ces objets peuvent être identifiés *localement*, directement dans le graphe $G$. Cette observation est le principal résultat du chapitre. Nous donnerons pour cela deux caractérisations des cliques maximales potentielles. La première est en forte relation avec les séparateurs minimaux du graphe. La deuxième est plus simple et n'utilise pas explicitement la notion de séparateur minimal. Les deux caractérisations nous permettront plus tard d'identifier les cliques maximales potentielles de toutes les classes de graphes citées dans le tableau 1.1 et aussi des graphes faiblement triangulés, pour lesquels le problème de la largeur arborescente et de la complétion minimale était ouvert. Le résultat algorithmique principal du chapitre 3 est que, étant donné un ensemble de sommets $K$ d'un graphe $G$, nous pouvons vérifier en temps polynomial si $K$ est une clique maximale potentielle de $G$, i.e. s'il existe une triangulation minimale $H$ de $G$ dans laquelle $K$ soit une clique maximale.

Au chapitre 4 nous nous intéressons à l'utilisation des cliques maximales potentielles d'un graphe pour calculer la largeur arborescente et la complétion minimale du graphe respectif. Nous donnons d'abord un algorithme simple qui, en partant des cliques maximales potentielles d'un graphe $G$, vérifie en temps polynomial si la largeur arborescente de $G$ est plus petite qu'une certaine valeur $k$. Nous obtenons un algorithme dont la complexité ne dépend pas de $k$, facilement transformable en un algorithme polynomial de calcul de la largeur arborescente. Son fonctionnement est en deux étapes. La première consiste à identifier toutes les cliques potentielles qui interviennent dans des triangulations de largeur arborescente au plus $k$. Si l'ensemble de ces cliques potentielles n'est pas vide, la deuxième étape de l'algorithme donne une décomposition arborescente de largeur au plus $k$ du graphe en entrée. Il est à noter que toute triangulation minimale de ce type peut être obtenue par l'algorithme. Nous donnons ensuite un algorithme qui, prenant en entrée les cliques maximales potentielles d'un graphe, calcule en temps polynomial la largeur arborescente et la complétion minimale du graphe. Cet algorithme, légèrement plus lent, utilise des techniques de programmation dynamique et n'est pas sans rappeler certains algorithmes de calcul de la largeur arborescente pour des classes particulières.

Le chapitre 5 présente les cliques maximales potentielles de toutes les classes de graphes déjà citées, pour lesquelles les problèmes de la largeur arborescente et de la complétion minimale ont été résolus. Nous unifions donc de cette manière tous ces algorithmes. Enfin, nous donnons une procédure calculant en temps polynomial les cliques maximales potentielles des graphes faiblement triangulés, dont il était connu qu'ils ont un nombre polynomial de séparateurs minimaux, sans que pour autant il existe d'algorithme calculant leur largeur arborescente et leur complétion minimale. Cette procédure nous permet donc de régler une question ouverte - le problème de la largeur arborescente et de la complétion minimale est polynomial pour les graphes faiblement triangulés. Le procédé que nous utilisons, très différent de ceux employés pour d'autres classes de graphes, nous laisse penser qu'il pourrait être généralisé à tous les graphes ayant un nombre polynomial de séparateurs. Le chapitre finit sur une conjecture qui permettrait de clore la conjecture d'ESA '93, sur le calcul de la largeur arborescente et de la complétion minimale des graphes avec "peu" de séparateurs minimaux.

Le chapitre 6 conclut cette thèse en proposant des extensions.

# Chapitre 2

# Préliminaires

Dans ce chapitre nous commençons par introduire les notions de base sur les graphes, les classes de graphes et les paramètres de triangulation qui nous concernent. Nous donnons ensuite les principaux résultats déjà connus, liés aux triangulations et au calcul de la largeur arborescente et de la complétion minimale des graphes.

## 2.1 Terminologie

Tout au long de la thèse nous considérerons des graphes simples, finis et non orientés. De plus, nos graphes seront connexes, car tous les paramètres qui nous intéressent (largeur arborescente, complétion minimale) se calculent pour les graphes non connexes en traitant séparément chacune des composantes connexes.

### 2.1.1 Graphes quelconques

Un graphe sera noté $G = (V, E)$, où $V$ est l'ensemble de sommets et $E$ est l'ensemble d'arêtes de $G$. Le nombre de sommets, respectivement d'arêtes sera noté $n$, respectivement $m$. Nous noterons simplement $xy$ une arête $\{x, y\}$ de $G$.

Un graphe $H = (W, F)$ tel que $W \subseteq V$ et $F \subseteq \{xy \in E | x, y \in W\}$ est appelé *sous-graphe (partiel)* de $G$. Si $W \subseteq V$ est un ensemble de sommets de $G$, le *sous-graphe induit* par les sommets $W$ est $G[W] = (W, F)$, ou $F = \{xy \in E | x, y \in W\}$. Un graphe $H = (V, F)$ tel que $E \subseteq F$ est dit *sur-graphe* de $G$. Un graphe est *complet* si tout couple de sommets distincts sont adjacents. Un ensemble de sommets qui induit un sous-graphe complet est appelé *clique*. Une *clique maximale* sera une clique de $G$, maximale par inclusion. On note $\omega(G)$ le nombre maximum de sommets d'une clique de $G$.

Si $x$ est un sommet de $G$, on notera $N_G(x)$ (ou simplement $N(x)$) les sommets adjacents à $x$ dans $G$. Si $W \subset V$ est un ensemble de sommets, $N_G(W)$ désigne le *voisinage* de $W$, c'est-à-dire l'ensemble des sommets de $V \backslash W$ ayant au moins un voisin dans $W$. On notera par $\overline{G} = (V, \overline{E})$ le *graphe complémentaire* de $G$ : $xy \in \overline{E}$ ssi $x \neq y$ et $xy \notin E$. Si $K$ est un ensemble de sommets de $G$, le graphe $G[V \backslash K]$ sera noté simplement $G \backslash K$.

Une suite de sommets distincts $[x_1, \ldots, x_p]$ de $G$ est appelée *chaîne de $x_1$ à $x_p$* si $x_1x_2, \ldots, x_{p-1}x_p$ sont des arêtes de $G$. Si de plus $x_px_1$ est une arête, alors $[x_1, \ldots, x_p]$ s'appelle *cycle* de $G$. Dans les deux cas, on appelle *corde* une arête qui relie deux sommets non consécutifs de la chaîne ou du cycle. Lorsque la chaîne ou le cycle n'ont pas de corde on parle de *chaîne (cycle) sans corde* ou de *chaîne induite* ou *cycle induit*. Le graphe $G$ est dit *connexe* si tout couple de sommets est relié par une chaîne. Un ensemble de sommets qui induit un sous-graphe connexe de $G$ et qui est maximal par inclusion pour cette propriété s'appelle *composante connexe* de $G$. Un graphe connexe sans cycles est appelé *arbre*.

Nous utiliserons la notion de *treillis*, qui est un type particulier d'ordre. Un ordre $P = (V, \leq)$ est une relation binaire, réflexive, antisymétrique et transitive sur l'ensemble $V$. Deux éléments $x$ et $y$ d'un ordre possèdent un plus grand minorant commun (ou borne inférieure), noté $x \wedge y$ si pour tout $z$ tel que $z \leq x$ et $z \leq y$ on a aussi $z \leq x \wedge y$. De la même façon, $x$ et $y$ possèdent un plus petit majorant commun (ou borne supérieure) noté $x \vee y$ si $\forall z, (x \leq z)$ et $(y \leq z) \Rightarrow x \vee y \leq z$. Si elles existent, ces bornes sont uniques. Un ordre tel que tout couple de sommets possède une borne inférieure et une borne supérieure est appelé *treillis*.

### 2.1.2 Classes de graphes

Nous n'épuiserons pas dans cette section la description des classes de graphes que nous aurons à traiter par la suite. Mais comme nous rencontrerons dans cet ouvrage plusieurs classes de graphes d'intersection, nous en donnons ici le principe.

On considère une famille $\mathcal{F}$ d'ensembles non vides. On lui associe le *graphe d'intersection $G_{\mathcal{F}}$* de $\mathcal{F}$ obtenu comme suit : chaque ensemble de $\mathcal{F}$ est représenté par exactement un sommet de $G_{\mathcal{F}}$ et deux sommets de $G_{\mathcal{F}}$ sont adjacents si et seulement si les ensembles correspondants de $\mathcal{F}$ s'intersectent. On dira que la famille $\mathcal{F}$ est un *modèle d'intersection* de $G_{\mathcal{F}}$.

Quand la famille $\mathcal{F}$ est formée d'ensemble particuliers, on obtient des classes de graphes intéressantes.

Les *graphes d'intervalles* sont les graphes d'intersection des intervalles d'un ordre total (de la droite réelle, par exemple).

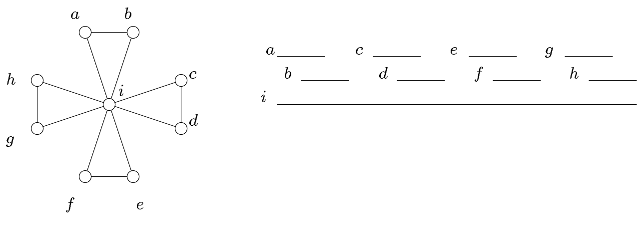


FIG. 2.1: Graphe d'intervalles

Les *graphes de cordes* sont obtenus comme intersections des cordes d'un cercle. Lorsque la famille $\mathcal{F}$ est formée d'arcs de cercle, on obtient les *graphes d'intervalles* *circulaires* . Ces deux dernières classes sont incluses dans la classe des graphes obtenus en intersectant des régions d'un cercle, où par région on entend la partie entre deux cordes qui ne se croisent pas.

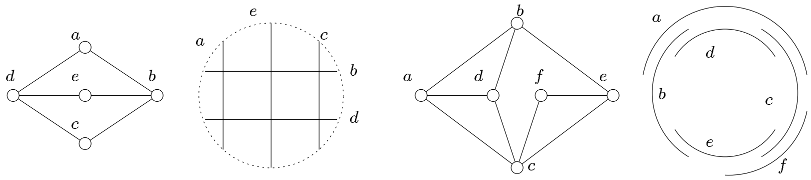


FIG. 2.2: Graphes de cordes et d'intervalles circulaires

Les *graphes triangulés* , qui s'obtiennent aussi par un modèle d'intersection, sont au cœur de cette thèse. Nous leur consacrons la section suivante.

## 2.2 Graphes triangulés

Il est difficile de savoir à quelle époque remonte la définition des graphes triangulés ; ces graphes apparaissent déjà en 1958 dans les travaux de Hajós.

**Définition 2.1** *Un graphe est dit* triangulé *si tout cycle ayant plus de trois sommets possède une corde.*

Dans la littérature anglo-saxonne, on trouve les dénominations de *triangulated graphs*, *chordal graphs* ou encore *rigid circuit graphs*.

On voit facilement que cette classe de graphes est héréditaire :

**Proposition 2.2** *Tous sous-graphe induit d'un graphe triangulé est triangulé.*

Cette classe de graphes a été intensément étudiée dans la littérature et connaît un certain nombre de caractérisations. La brève présentation qui suit est basée notamment sur le livre de Golumbic [Gol80] et un article de Blair et Peyton [BP93].

Le plus important pour la suite est de voir ces graphes comme des "arbres généralisés". Notons $\mathcal{K}_G = \{\Omega_1, \ldots, \Omega_p\}$ l'ensemble des cliques maximales de $G$. Soit $\mathcal{T}$ un arbre sur $\mathcal{K}_G$, i.e. chaque clique de $\mathcal{K}_G$ correspond à un noeud de $\mathcal{T}$. Nous dirons aussi que les sommets de $\mathcal{T}$ sont étiquetés par les cliques maximales de $G$ et nous désignerons simplement par $\Omega$ le noeud de $\mathcal{T}$ étiqueté par la clique maximale $\Omega$. Nous dirons que $\mathcal{T}$ est un *arbre de cliques* de $G$ s'il satisfait la *propriété d'intersection des cliques* : pour tout couple de cliques distinctes $\Omega$ et $\Omega'$, l'ensemble $\Omega \cap \Omega'$ est contenu dans toutes les cliques sur la chaîne reliant $\Omega$ et $\Omega'$ dans l'arbre $\mathcal{T}$.

Le théorème suivant est dû à Gavril [Gav74] :

**Théorème 2.3 ([Gav74])** *Un graphe $G$ est triangulé si et seulement si il admet un arbre de cliques.*

La figure 2.3 $a, b$ montre un exemple de graphe triangulé, avec un de ses arbres de cliques.

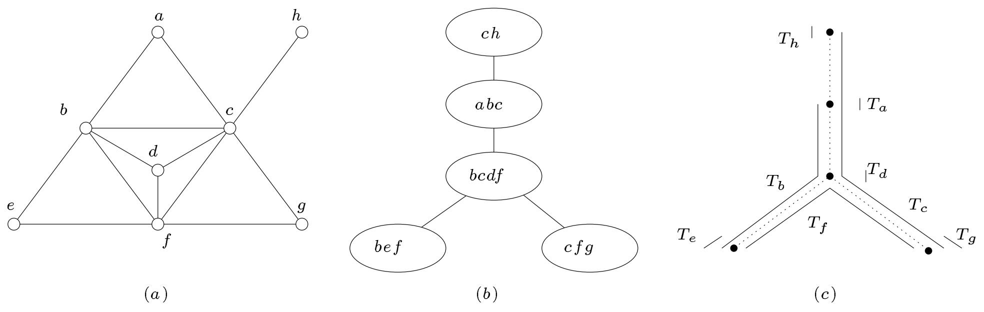


FIG. 2.3: Graphe triangulé et représentation arborescentes

Notons qu'à un graphe triangulé $G$ on peut en général associer plusieurs arbres de cliques, non isomorphes. On trouve par exemple dans Paul [Pau98] une étude sur la formation de tous ces arbres et des relations entre eux.

D'un point de vue technique, nous utiliserons beaucoup la notion d'arbre de cliques. Mais en fait, cette notion est fortement liée à la représentation des graphes triangulés comme des graphes modélisant l'intersection des sous-arbres d'un arbre. Cette dernière a l'avantage d'être plus intuitive.

**Théorème 2.4 ([Gav74])** *Un graphe est triangulé si et seulement si il est le graphe d'intersection d'une famille de sous-arbres d'un arbre.*

Il est facile de voir que le graphe d'intersection des sous-arbres d'un arbre est triangulé. La réciproque est plus technique, voir par exemple [Gol80] pour une preuve. Mais remarquons qu'un arbre de cliques $\mathcal{T}$ de $G$ fournit directement le modèle d'intersection de $G$ : pour tout $x \in V(G)$, on désigne par $T_x$ l'ensemble des noeuds de $\mathcal{T}$ dont l'étiquette contient $x$. Par la propriété d'intersection des cliques, $T_x$ induit dans $\mathcal{T}$ un sous-arbre. La famille de ces sous-arbres est un modèle d'intersection de $G$, comme on peut le voir sur la figure 2.3 $c$. Réciproquement, à partir d'un modèle d'intersection de $G$ on obtient assez facilement un arbre de cliques.

Une dernière caractérisation des graphes triangulés est donnée par les *ordres d'élimination simplicielle.*

**Définition 2.5** *Soit $G = (V, E)$ un graphe. Un sommet $x$ est dit* simpliciel *si son voisinage $N_G(x)$ est une clique. Un ordre $x_1, \dots x_n$ sur les sommets de $G$ tel que pour tout $i, 1 \leq i \leq n$ le sommet $x_i$ est simpliciel dans $G[\{x_i, \dots, x_n\}]$ est appelé* ordre d'élimination simplicielle.

La preuve de ce théorème se trouve par exemple dans [Gol80].

**Théorème 2.6** *Un graphe $G$ est triangulé si et seulement si il admet un ordre d'élimination simplicielle.*

Toujours pour le graphe de la figure 2.3, un ordre d'élimination simplicielle est par exemple $h, a, e, g, d, b, c, f$.

Cette propriété est à la base des algorithmes linéaires de reconnaissance des graphes triangulés tels que LexBFS et MCS, qui fonctionnent, d'ailleurs, de manière assez surprenante.

## 2.3 Problèmes de triangulation

Dans cette partie nous définirons la largeur arborescente et la complétion minimale d'un graphe, dont le calcul est exprimable comme un "problème de triangulation". Le calcul de ces paramètres pour un graphe $G$ reviendra à trouver un "bon" surgraphe triangulé $H$ de $G$.

**Définition 2.7** *Soit $G = (V, E)$ un graphe quelconque. On dit que le graphe $H = (W, F)$ est une* triangulation *de $G$ si $V = W$, $E \subseteq F$ et $H$ est un graphe triangulé. En d'autres termes, $H$ est un sur-graphe triangulé de $G$.*

*On dit que $H = (V, F)$ est une* triangulation minimale *de $G$ si $H$ est une triangulation de $G$ et pour tout $F'$ tel que $E \subseteq F' \subset F$, le graphe $H' = (V, F')$ n'est pas une triangulation de $G$.*

Rose, Tarjan et Lueker donnent une caractérisation simple des triangulations minimales [RTL76] :

**Lemme 2.8** *Soit $H$ une triangulation d'un graphe $G$. $H$ est une triangulation minimale de $G$ si et seulement si, pour toute arête $e$ de $E(H)\backslash E(G)$, le graphe $H - e = (V, E(H)\backslash\{e\})$ n'est pas une triangulation de $G$.*

Nous pouvons définir maintenant les paramètres de graphes dont le calcul fait l'objet de cette thèse :

**Définition 2.9** *Une* décomposition arborescente *d'un graphe $G = (V, E)$ est un couple $(\{X_i, i \in I\}, \mathcal{T} = (I, E_T))$ où les $X_i$ sont des ensembles de sommets de $G$ et $\mathcal{T}$ est un arbre tel que :*

1. $\bigcup_{i \in I} X_i = V$,
2. *pour chaque arête $xy$ de $G$, il existe un $i \in I$ tel que $x, y \in X_i$,*
3. *pour tout $x \in V$, l'ensemble $T_x$ des sommets $i \in I$ de $\mathcal{T}$ tels que $x \in X_i$ induit dans $\mathcal{T}$ un sous-arbre.*

*La* largeur *d'une décomposition $(\{X_i, i \in I\}, \mathcal{T} = (I, E_T))$ est* $\max_{i \in I} |X_i| - 1$. *La* largeur arborescente *(*treewidth*, en anglais) de $G$, notée* $\mathrm{tw}(G)$, *est la largeur minimum de toutes les décompositions arborescentes de $G$.*

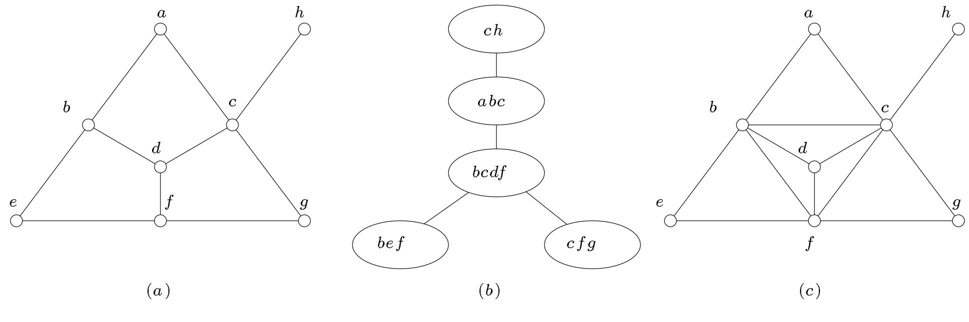


FIG. 2.4: Décomposition arborescente et triangulation associée

La figure 2.4 $a, b$ montre un graphe et une de ses décompositions arborescentes, de largeur 3. En l'occurrence, c'est aussi la largeur arborescente de ce graphe. Notons qu'il est facile de trouver une décomposition arborescente d'un graphe : on peut toujours prendre un arbre avec un seul noeud et lui mettre comme étiquette l'ensemble $V$. La largeur de cette décomposition est $n-1$, si $n$ est le nombre de sommets de $V$.

Regardons la largeur arborescente de quelques graphes simples. La largeur arborescente d'une clique est $n-1$, c'est donc "le pire cas" car on a toujours $\mathrm{tw}(G) \leq n-1$. Il est assez facile de voir que les graphes de largeur arborescente 1 sont exactement les forêts - ou les arbres, si l'on se restreint aux graphes connexes. Ceci est censé justifier la constante "-1" de la définition de la largeur arborescente. On peut voir aussi que la largeur arborescente des cycles est 2. Plus difficile, la largeur arborescente d'une grille de taille $a \times b$ est $\min(a, b)$.

On observe facilement que la définition d'une décomposition arborescente fait intervenir une famille de sous-arbres d'un arbre dont le graphe d'intersection est un sur-graphe de $G$. Nous allons redéfinir la largeur arborescente de $G$ en passant par ses triangulations minimales, et c'est cette version de la définition de la largeur arborescente que nous allons utiliser par la suite.

Considérons une décomposition arborescente $(\{X_i, i \in I\}, \mathcal{T})$ du graphe $G$. Par définition, les ensembles $T_x = \{i \in I | x \in X_i\}$ forment des sous-arbres de $\mathcal{T}$. Considérons maintenant le graphe $H = (V, F)$ où $xy \in F$ si et seulement si $x$ et $y$ sont dans une même étiquette $X_i$. Clairement la famille $\{T_x, x \in V\}$ de sous-arbres de $\mathcal{T}$ est un modèle d'intersection de $H$, et par le théorème 2.4 on obtient que $H$ est un graphe triangulé. Par la deuxième condition de la définition 2.9, toute arête de $G$ est aussi une arête de $H$. En conséquence, $H$ est une triangulation de $G$. Réciproquement, tout modèle d'intersection d'une triangulation $H$ de $G$ est clairement une décomposition arborescente de $G$.

Montrons que la largeur de la décomposition arborescente $(\{X_i, i \in I\}, \mathcal{T})$ est exactement $\omega(H) - 1$. Tout d'abord, les sommets de chaque étiquette $X_i$ forment une clique dans $H$, donc la largeur de la décomposition ne dépasse pas $\omega(H) - 1$. Il reste à montrer que pour toute clique $\Omega$ de $H$, il existe une étiquette $X_i$ contenant $\Omega$. Utilisons directement la propriété de Helly des sous-arbres d'un arbre (voir [Gol80] pour une preuve) :

**Lemme 2.10** *Une famille $\{T_x | x \in V\}$ de sous-arbres d'un arbre satisfait la* propriété de Helly, *c'est-à-dire que pour toute sous-famille $W \subseteq V$ satisfaisant $T_x \cap T_y \neq \emptyset$ pour tout $x, y \in W$, on a aussi $\cap_{x \in W} T_x \neq \emptyset$.*

Si $\Omega$ est une clique de $H$, les sous-arbres $\{T_x | x \in \Omega\}$ s'intersectent deux à deux. En conséquence, il existe un $i \in I$ tel que $i \in \cap_{x \in \Omega} T_x$. On en conclut que $\Omega \subseteq X_i$. Il reste que $\max\{|X_i|, i \in I\} = \omega(H)$.

Nous avons prouvé :

**Proposition 2.11** *La largeur arborescente de $G$ est égal au minimum, pour toutes les triangulations $H$ de $G$, de $\omega(H) - 1$.*

C'est donc cette définition de la largeur arborescente que nous utiliserons pour la calculer.

Le deuxième paramètre que nous allons traiter se définit directement à partir des triangulations minimales, comme étant le nombre minimum d'arêtes que l'on doit rajouter à un graphe $G$ pour obtenir un sur-graphe triangulé.

**Définition 2.12** *Soit $G = (V, E)$ un graphe. La* complétion minimale *(en anglais :* minimum fill-in*) de $G$, notée* $\mathrm{cm}(G)$ *est* $\min\{|F| - |E|$*, où* $H = (V, F)$ *est une triangulation de* $G\}$.

Pour calculer la largeur arborescente ou la complétion minimale d'un graphe $G$ il faut donc chercher des triangulations de $G$, tout en minimisant la taille de la clique maximum dans le premier cas, le nombre d'arêtes dans le deuxième. On peut donc se limiter à chercher parmi les *triangulations minimales* de $G$. C'est pourquoi les algorithmes que nous présenterons par la suite calculent de façon presque identique les deux paramètres et seront basés sur des caractérisations "algorithmiques" des triangulations minimales d'un graphe.

## 2.4 Séparateurs minimaux

Les algorithmes de calcul de la largeur arborescente et de la complétion minimale passent presque tous par la notion de *séparateur minimal.* Nous définissons ici ces objets et nous étudions en détail les séparateurs minimaux des graphes triangulés. Nous montrons ensuite comment énumérer tous les séparateurs minimaux d'un graphe.

### 2.4.1 Généralités sur les séparateurs minimaux

**Définition 2.13** *Soit $G = (V, E)$ un graphe et soient $a$ et $b$ deux sommets de $G$. Un ensemble de sommets $S \subseteq V$ est appelé* $a, b$-séparateur *(ou simplement* séparateur*) si $a$ et $b$ se trouvent dans des composantes connexes différentes de $G \backslash S$. Un $a, b$-séparateur est dit* $a, b$-séparateur minimal *s'il ne contient pas un autre $a, b$-séparateur. L'ensemble de sommets $S \subseteq V$ est appelé* séparateur minimal *de $G$ s'il existe un couple de sommets $a, b$ tel que $S$ soit un $a, b$-séparateur minimal.*

Remarquons qu'un séparateur minimal peut être strictement contenu dans un autre séparateur minimal, voir figure 2.5 où le $h, g$-séparateur minimal $\{c\}$ est strictement inclus dans le $a, g$-séparateur minimal $\{c, f\}$.

Si $S$ est un ensemble de sommets de $G$ (pas forcément un séparateur), nous noterons $\mathcal{C}_G(S)$ l'ensemble des composantes connexes de $G \backslash S$. Soit $C \in \mathcal{C}_G(S)$ une telle composante connexe. Si tout sommet de $S$ a au moins un voisin dans $C$, nous dirons que $C$ est une *composante connexe pleine associée à $S$* ou que $C$ est une *composante connexe pleine induite par $S$* dans $G$. L'ensemble des composantes connexes pleines associées à $S$ sera noté $\mathcal{C}^*_G(S)$. Sur la figure 2.5, les composantes connexes de $G \backslash T$ sont $\{a, b, e, d\}, \{g\}$ et $\{h\}$, et les deux premières sont pleines par rapport à $T$. Pour un sommet $a \in V \backslash S$ de $G$, nous noterons $C^a_G(S)$ la composante connexe de $G \backslash S$ contenant $a$.

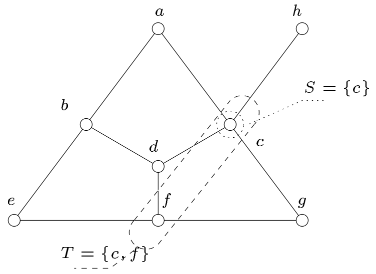


FIG. 2.5: Séparateurs minimaux

**Proposition 2.14** *Soient $S \subset V$ un ensemble de sommets et $C, D$ deux composantes connexes différentes de $G \backslash S$. Les propositions suivantes sont équivalentes :*

1. *$C$ et $D$ sont des composantes connexes pleines associées à $S$.*
2. *Il existe $a \in C$ et $b \in D$ tels que $S$ soit un $a, b$-séparateur minimal.*
3. *Pour tout $a \in C$ et $b \in D$, $S$ est un $a, b$-séparateur minimal.*

*Preuve.* Montrons que (1) $\Rightarrow$ (3). Soient $a$ et $b$ des sommets quelconques des composantes connexes pleines $C$, respectivement $D$ associées à $S$. $S$ est clairement un $a, b$-séparateur. Supposons qu'il existe un $a, b$-séparateur $S' \subset S$. Soit $x \in S \backslash S'$. Par définition d'une composante connexe pleine, tout sommet de $S$ a au moins un voisin dans $C$, respectivement $D$. Il existe alors une chaîne de $x$ à $a$ dans $C \cup \{x\}$ et une chaîne de $x$ à $b$ dans $D \cup \{x\}$. On en conclut que $a$ et $b$ sont reliés dans $G \backslash S'$, contredisant le fait que $S'$ est un $a, b$-séparateur. Nous en déduisons que $S$ est un $a, b$-séparateur minimal.

Montrons maintenant que (2) $\Rightarrow$ (1). Supposons que la composante connexe $C$ de $G \backslash S$ ne soit pas pleine par rapport à $S$ et soit $S' \subset S$ l'ensemble de sommets de $S$ ayant au moins un voisin dans $C$. Montrons que $S'$ est un $a, b$-séparateur. Comme $C$ est une composante connexe de $G \backslash S$, on a $N_G(C) \subseteq S$. Par construction de $S'$, on a même $N_G(C) \subseteq S'$. Donc toute chaîne entre un sommet de $C$ et un sommet de $G \backslash (C \cup S')$ passe par au moins un sommet de $S'$. On conclut que $a$ et $b$ sont dans des composantes connexes différentes de $G \backslash S'$.

L'implication (3) $\Rightarrow$ (2) est bien entendu élémentaire. $\diamond$

On en déduit aisément :

**Corollaire 2.15** *$S$ est un séparateur minimal de $G$ si et seulement si $S$ a au moins deux composantes pleines.*

La petite proposition qui suit donne une explication sur l'existence des séparateurs minimaux qui ne sont pas minimaux par inclusion. Ceux-ci sont exactement les séparateurs minimaux induisant aussi des composantes non-pleines :

**Proposition 2.16** *Soient $G$ un graphe, $S$ un séparateur minimal de $G$ et $C$ une composante connexe de $G \backslash S$, non-pleine par rapport à $S$. Alors l'ensemble de sommets $S' \subseteq S$ ayant au moins un voisin dans $C$ est un séparateur minimal de $G$.*

*Preuve.* Comme $S$ est un séparateur minimal, il induit une composante connexe pleine $D$ dans $G$. De la même façon que dans la preuve de la proposition précédente, $C$ est une composante pleine associée à $S'$. D'un autre côté, $D$ est inclus dans une autre composante connexe, toujours pleine, associée à $S'$. Par le corollaire précédent, $S'$ est un séparateur minimal de $G$. $\diamond$

Remarquons que l'on peut regarder un $a, b$-séparateur $S$ comme un ensemble de sommets tel que toute chaîne de $a$ à $b$ passe par $S$. De plus $S$ est un $a, b$-séparateur minimal si et seulement si pour tout $x \in S$, il existe une chaîne de $a$ à $b$ qui intersecte $S$ uniquement en $x$.

### 2.4.2 Séparateurs minimaux des graphes triangulés

Dirac a donné en 1961 une première caractérisation des graphes triangulés à l'aide des séparateurs minimaux [Dir61] :

**Théorème 2.17** *Un graphe $G$ est triangulé si et seulement si tous ses séparateurs minimaux sont des cliques.*

*Preuve.* Montrons déjà que tous les séparateurs minimaux d'un graphe triangulé $G$ sont des cliques. Considérons deux sommets $x$ et $y$ de $G$ appartenant à un même séparateur minimal $S$. Par la proposition 2.14, $S$ induit dans $G$ au moins deux composantes pleines $C$ et $D$. Comme $C$ est une composante connexe pleine associée à $S$, nous pouvons prendre deux sommets $c_1$ et $c_p$ voisins de $x$, respectivement de $y$ dans $C$. Les sommets $c_1$ et $c_2$ sont reliés par une chaîne contenue dans la composante connexe $C$. Il existe alors une chaîne $[x, c_1, \ldots, c_p, y]$ de $x$ à $y$ telle que tous les $c_i$ sont contenus dans $C$. Nous pouvons prendre cette chaîne sans corde, excepté éventuellement l'arête $xy$. De la même façon, soit $[x, d_1, \ldots d_q, y]$ une chaîne sans corde reliant $x$ à $y$, telle que tous les $d_i$ sont dans $D$. Comme $S$ sépare les composantes $C$ et $D$, un sommet $c_i$ ne peut pas être adjacent à un sommet $d_j$. Le cycle $[x, c_1, \ldots c_p, y, d_q, \ldots, d_1]$ a clairement au moins quatre sommets. Sa seule corde possible est $xy$, donc $xy \in E(G)$.

Réciproquement, soit $G$ un graphe triangulé. Supposons que $G$ ait un cycle sans corde $[x_1, x_2, \ldots, x_p]$ de longueur au moins 4. Montrons que $G$ a un séparateur minimal qui n'induit pas une clique. Nous allons construire un $x_1, x_3$-séparateur minimal contenu dans le voisinage de $x_1$. Puisque $N_G(x_1)$ est un $x_1, x_3$-séparateur nous pouvons en extraire un $x_1, x_3$-séparateur minimal $S$. Il est clair que $S$ contient $x_2$. Montrons que $S$ contient $x_p$. La chaîne $[x_4, \ldots, x_p]$ (éventuellement $p = 4$) intersecte forcément $S$. Si cette intersection a lieu en un sommet $x_l$ autre que $x_p$ alors comme $x_l \in N(x_1)$ le cycle initial aurait une corde, à savoir $x_1 x_l$. Le séparateur $S$ est donc un $x_1 x_3$ séparateur minimal qui n'induit pas une clique car l'arête $x_2 x_p$ n'existe pas. $\diamond$

En fait, le $x_1, x_3$-séparateur minimal contenu dans le voisinage de $x_1$ est unique, comme le montre la proposition suivante.

**Proposition 2.18** *Pour tout couple $(a, b)$ de sommets non adjacents il existe un unique $a, b$-séparateur minimal contenu dans $N(a)$.*

*Preuve.* Comme $N(a)$ sépare $a$ et $b$, nous savons que $N(a)$ contient au moins un $a,b$-séparateur minimal. Supposons qu'il existe deux $a,b$-séparateurs minimaux $S_1$ et $S_2$ contenus dans $N(a)$. Nous avons clairement $S_1 \backslash S_2 \neq \emptyset$ et $S_2 \backslash S_1 \neq \emptyset$. Soit $c \in S_2 \backslash S_1$, il existe une chaîne reliant $a$ et $b$ ne coupant $S_2$ qu'en $c$. Cette chaîne est de la forme $\mu = [a, x_1, \ldots, x_k, b]$ avec $x_1 = c$ ; si ce n'est pas le cas, c'est-à-dire si $c = x_l$ avec $1 < l \leq k$, on prend la sous-chaîne $[a, c, x_{l+1}, \ldots, b]$. $S_1$ étant un $a,b$-séparateur minimal, $\mu$ intersecte $S_1$ en un sommet $x_p \in S_1 \backslash S_2$. La sous-chaîne $[a, x_p, \ldots, x_k, b]$ relie $a$ à $b$ sans couper $S_2$, contredisant le fait que $S_2$ soit un $a,b$-séparateur. $\diamond$

**Remarque 2.19** *Nous pouvons dire que le séparateur minimal $S$ est* le plus proche *de $a$ parmi les $a,b$-séparateurs minimaux ; nous retrouverons ce séparateur un peu plus loin.*

Le théorème de Dirac est expliqué de manière beaucoup plus structurelle par le très important théorème qui suit, qui se trouve dans les travaux de Ho et Lee [HL89] et Lundquist [Lun90] :

**Théorème 2.20** *Soit $G$ un graphe triangulé et soit $\mathcal{T}$ un arbre de cliques* quelconque *de $G$. Un ensemble de sommets $S$ est un séparateur minimal de $G$ si et seulement si $S$ s'écrit sous la forme $\Omega \cap \Omega'$, où $\Omega$ et $\Omega'$ sont des cliques maximales de $G$, adjacentes dans l'arbre de cliques $\mathcal{T}$.*

Avant d'aborder la preuve, donnons un lemme simple sur la structure d'un arbre de cliques. Si $\mathcal{T}$ est un arbre de cliques d'un graphe triangulé $G$ et $\Omega$ et $\Omega'$ sont deux cliques adjacentes dans $\mathcal{T}$, nous allons noter $\mathcal{T}_\Omega$ et $\mathcal{T}_{\Omega'}$ les sous-arbres de $\mathcal{T}$ obtenus en enlevant l'arête $\Omega\Omega'$, où $\Omega \in \mathcal{T}_\Omega$ et $\Omega' \in \mathcal{T}_{\Omega'}$. Posons $V_\Omega$ et $V_{\Omega'}$ les ensembles de sommets de $G$ qui appartiennent à au moins une étiquette de $\mathcal{T}_\Omega$, respectivement $\mathcal{T}_{\Omega'}$ :

**Lemme 2.21** *$S = \Omega \cap \Omega'$ sépare dans $G$ tout sommet $a \in V_\Omega \backslash S$ de tout sommet $b \in V_{\Omega'} \backslash S$.*

*Preuve.* Prenons une chaîne $\mu = [a = x_1, x_2, \ldots, x_k = b]$ de $G$ et supposons qu'elle n'intersecte pas $S$. Remarquons que $V_\Omega \cap V_{\Omega'} = S$, en effet si un sommet $x$ se trouve à la fois dans une étiquette de $\mathcal{T}_\Omega$ et de $\mathcal{T}_{\Omega'}$, comme les étiquettes de $\mathcal{T}$ contenant $x$ forment un sous-arbre de $\mathcal{T}$, $x$ doit se trouver dans $\Omega$ et $\Omega'$. Comme $a, b \notin \Omega \cap \Omega'$, nous avons que $a$ n'apparaît pas dans les étiquettes de $\mathcal{T}_{\Omega'}$ et $b$ n'apparaît pas dans celles de $\mathcal{T}_\Omega$. Soit donc $x_i$ le premier sommet de la chaîne $\mu$ qui se trouve dans une étiquette de $\mathcal{T}_{\Omega'}$. Comme $x_{i-1}$ ne se trouve pas parmi les étiquettes de $\mathcal{T}_{\Omega'}$ et que $x_{i-1}$ est adjacent à $x_i$, il existe une clique maximale qui contient $x_i$ et $x_{i-1}$ et donc les deux sont forcément contenus dans une étiquette de $\mathcal{T}_\Omega$. On en déduit que $x_i$ se trouve à la fois dans les étiquettes de $\mathcal{T}_\Omega$ et dans $\mathcal{T}_{\Omega'}$, c'est-à-dire $x_i \in \Omega \cap \Omega' = S$. On en conclut que $S$ sépare $a$ et $b$. $\diamond$

Nous pouvons revenir maintenant à la preuve du théorème 2.20.

*Preuve.* Montrons d'abord que si $\Omega$ et $\Omega'$ sont deux cliques adjacentes dans $\mathcal{T}$, alors $S = \Omega \cap \Omega'$ est un séparateur minimal. Plus précisément, nous considérons deux sommets $a \in \Omega \backslash \Omega'$ et $b \in \Omega' \backslash \Omega$ et nous prouvons que $S$ est un $a,b$-séparateur minimal. $S$ est un $a,b$-séparateur par le lemme 2.21. Soient $C^a$ et $C^b$ les composantes connexes de $G\backslash S$ contenant $a$, respectivement $b$. Comme $S \subseteq N(a)$ et $S \subseteq N(b)$, les composantes $C^a$ et $C^b$ sont des composantes pleines associées à $S$ et donc $S$ est un $a,b$-séparateur minimal par la proposition 2.14.

Réciproquement, montrons que pour tout $a,b$-séparateur minimal $S$, et pour tout arbre de cliques $\mathcal{T}$, il existe deux cliques $\Omega$ et $\Omega'$ adjacentes dans $\mathcal{T}$ telles que $S = \Omega \cap \Omega'$. Soient $T_a$, respectivement $T_b$ les sous-arbres de $\mathcal{T}$ induits par les étiquettes contenant $a$, respectivement $b$. Clairement $T_a \cap T_b = \emptyset$. On choisit $\Omega_a \in T_a$ et $\Omega_b \in T_b$ de sorte que la chaîne de $\Omega_a$ à $\Omega_b$ dans $\mathcal{T}$ soit de longueur minimum. Soient $[\Omega_a = \Omega_1, \ldots, \Omega_p = \Omega_b]$ les cliques de cette chaîne. Notons $S_i = \Omega_i \cap \Omega_{i+1}$ et soit $\mathcal{S} = \{S_1, \ldots, S_{p-1}\}$. Nous allons montrer que que $S \in \mathcal{S}$.

Commençons par prouver qu'il existe au moins un $S_i \in \mathcal{S}$ tel que $S_i \subseteq S$. Si ce n'était pas le cas, pour tout $i, 1 \leq i \leq p-1$ il existerait un sommet $x_i \in S_i \backslash S$. Comme $x_i, x_{i+1} \in \Omega_{i+1}$ pour tout $i$, nous en déduisons que $[a, x_1, \ldots, x_{p-1}, b]$ est une chaîne, qu'elle relie $a$ et $b$ et qu'elle n'intersecte pas $S$. Ceci contredit le fait que $S$ soit un $a,b$-séparateur. Il s'ensuit que l'on peut choisir un $S_i$ contenu dans $S$. Mais on sait que $S_i$ sépare $a$ et $b$ dans $G$ par le lemme 2.21. Puisque $S$ est un $a,b$-séparateur minimal et qu'il contient $S_i$, on en déduit que $S = S_i$, ce qui achève la preuve. $\diamond$

Nous avons vu donc que les séparateurs minimaux d'un graphe triangulé $G$ sont exactement les intersections des cliques adjacentes de *n'importe quel arbre de cliques* de $G$. Il est naturel d'étiqueter les arêtes $\Omega\Omega'$ d'un arbre de cliques $\mathcal{T}$ par le séparateur minimal $S = \Omega \cap \Omega'$. Il a été prouvé par Ho et Lee [HL89] et Lundquist [Lun90] qu'en fait le multi-ensemble formé par les étiquettes des arêtes d'un arbre de cliques est le même, indépendemment de l'arbre choisi.

Dans un arbre de cliques $\mathcal{T}$ quelconque, considérons un séparateur minimal $S$ inclus dans une clique maximale $\Omega$. Nous savons que $S$ s'écrit sous la forme $\Omega_1 \cap \Omega_2$ pour une certaine paire de cliques adjacentes $\Omega_1$ et $\Omega_2$. Par contre il n'existe pas forcément une clique $\Omega'$ adjacente à $\Omega$ dans $\mathcal{T}$ telle que $\Omega \cap \Omega'$. Toutefois, pour chaque clique maximale $\Omega$ on peut trouver un arbre de cliques tel que tout séparateur $S \subset \Omega$ soit sous cette forme $\Omega \cap \Omega'$, comme nous allons le voir plus tard.

### 2.4.3 Énumération des séparateurs minimaux d'un graphe

Tout le travail de cette thèse est lié à la compréhension de la structure des séparateurs minimaux des graphes. On notera $\Delta_G$ l'ensemble des séparateurs minimaux du graphe $G$. Comme nous serons amenés à utiliser plus tard *tous* les séparateurs minimaux d'un graphe, nous esquissons ici l'algorithme de Kloks et Kratsch [KK94] qui les énumère en un temps polynômial en la taille du graphe et du nombre de ses séparateurs minimaux. Cet algorithme procède en énumérant les $a,b$-séparateurs minimaux pour tout couple de sommets $a$ et $b$. Hélas nous ne disposons d'aucun algorithme d'énumération qui travaille directement sur la totalité des séparateurs minimaux d'un graphe. Ceci vient probablement du fait que nous ne savons pas munir $\Delta_G$ d'une bonne structure, par exemple d'une structure d'ordre. La situation est complètement différente si l'on considère seulement les $a,b$-séparateurs minimaux de $G$, pour un couple de sommets $a$ et $b$ fixés. Cet ensemble - notons-le $\Delta_G(a,b)$ - peut être muni de façon très naturelle d'une structure de treillis. Voici les principaux résultats de [KK94, KK98], donnés sans preuve, mais dont il est facile de se convaincre.

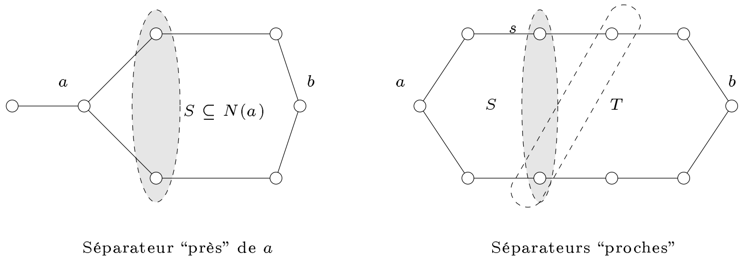


FIG. 2.6: Structure des $a,b$-séparateurs minimaux

Comme nous l'avons fait remarquer (proposition 2.18 et remarque 2.19), il existe un unique $a,b$-séparateur minimal qui est *le plus proche de* $a$, c'est-à-dire qui est contenu dans $N_G(a)$. Il s'agit du séparateur formé par les sommets de $N_G(a)$ ayant au moins un voisin dans la composante connexe de $G \backslash N_G(a)$ contenant $b$. Considérons maintenant un $a,b$ séparateur minimal $S$, qui n'est pas le plus proche de $b$. Il existe alors $s \in S$ non adjacent à $b$. Soit $T'$ le $s,b$-séparateur minimal le plus proche de $s$ dans le graphe $G' = G[C^b_G(S) \cup \{s\}]$ et posons $S'$ l'ensemble de sommets de $S$ n'ayant pas de voisin dans $G^b_{G'}(T')$. Kloks et Kratsch prouvent dans [KK94] que l'ensemble de sommets $T = S \cup T' \backslash S'$ est un $a,b$-séparateur minimal de $G$. En quelque sorte, $T$ est un $a,b$-séparateur minimal "proche de S". De cette manière, en partant du $a,b$-séparateur minimal le plus proche de $a$ et en allant de proche en proche, on obtient tous les $a,b$-séparateurs minimaux de $G$. Comme toutes ces résultats sont constructifs, on trouve :

**Théorème 2.22 ([KK94])** *L'ensemble $\Delta_G$ des séparateurs minimaux de $G$ peut être calculé en temps $\mathcal{O}(n^5|\Delta_G|)$.*

Cet algorithme, ainsi que des versions ultérieures [KK98, SL97], calculent tous les séparateurs minimaux d'un graphe en énumérant les $a,b$-séparateurs minimaux pour tout les couples de sommets $a$ et $b$. La proposition 2.14 suggère qu'un séparateur minimal peut être $a,b$-séparateur minimal pour un nombre important de couples $a,b$. On pourrait donc espérer un algorithme beaucoup plus performant d'énumération de tous les séparateurs minimaux d'un graphe si on arrivait à les considérer globalement, et non pas comme l'union des $a,b$-séparateurs minimaux pour toutes les paires $a,b$. Mais, comme nous l'avons dit, nous ne sommes pas arrivés à munir les séparateurs minimaux d'un graphe d'une relation dont on saurait maîtriser la structure, alors que l'on peut prouver qu'une version de la relation de $a,b$-séparateurs "proches" décrite plus haut induit une structure de treillis sur $\Delta_G(a,b)$.

Une tentative intéressante d'ordonner les séparateurs minimaux d'un graphe est due à Berry [Ber95]. Elle utilise une autre notion, de "séparateur deux-coloriable", dont les séparateurs minimaux sont un cas particulier. On obtient un treillis sur ces objets, qui est en fait le treillis de Galois de la relation "$x$ et $y$ sont des sommets distincts, non adjacents, de $G$". Le bon côté est que chaque treillis des $a, b$-séparateurs minimaux est un sous-ordre du treillis des séparateurs deux-coloriables. Le mauvais côté est que le nombre de séparateurs deux-coloriables peut être exponentiellement plus grand que celui des deux-séparateurs et surtout qu'un séparateur minimal peut être représenté par plusieurs séparateurs deux-coloriables, et c'est cette duplication de l'information qui engendre artificiellement la "bonne structuration".

## 2.5 Triangulations minimales et séparateurs

Dans cette section nous allons faire une première la relation entre les triangulations minimales et les séparateurs minimaux d'un graphe. Nous verrons que cette caractérisation des triangulations minimales ne fournit pas de solution algorithmique au calcul des paramètres qui nous intéressent.

Soient $S$ un séparateur minimal de $G$ et $C \in \mathcal{C}(S)$ une composante connexe de $G \backslash S$. Nous dirons que $(S, C) = S \cup C$ est un *bloc* de $S$ dans $G$. Si $C$ est une composante pleine associée à $S$, nous dirons que le bloc $(S, C)$ est *plein* . Notons que l'ensemble de sommets d'un bloc détermine de façon unique la paire $(S, C)$. En effet, si $K$ est l'ensemble de sommets du bloc, alors $S$ est formé exactement des sommets de $K$ ayant un voisin dans $V(G) \backslash K$.

**Définition 2.23** *Soient $S$ et $T$ deux séparateurs minimaux de $G$. On dit que $S$* croise *$T$, noté $S \sharp T$, si $S$ intersecte au moins deux composantes connexes de $G \backslash T$. Dans le cas contraire on dira que $S$ est* parallèle *à $T$, noté $S \| T$.*

Dans la thèse de Parra [Par96], le graphe $(\Delta_G, \|)$ de la relation de parallélisme entre séparateurs est appelé *graphe des séparateurs* de $G$.

**Lemme 2.24** *Les relations de parallélisme et de croisement entre séparateurs minimaux sont symétriques.*

*Preuve.* Montrons que si $S \sharp T$, alors $T \sharp S$. Soient $a$ et $b$ deux sommets de $S$ se trouvant dans des composantes connexes distinctes de $G \backslash T$. Par définition, $T$ est un $a, b$-séparateur. Soient $C$ et $D$ deux composantes pleines associées à $S$. Notons que $C$ et $D$ existent par la proposition 2.14. Soit maintenant une chaîne de $a$ à $b$ dans $G$, contenue dans $C \cup \{a, b\}$. Cette chaîne existe par définition d'une composante pleine. Puisque $T$ est un $a, b$-séparateur, $T$ doit intersecter la chaîne et en conséquence $T$ intersecte $C$. Pour les mêmes raisons, $T$ intersecte $D$ et donc $T \sharp S$. On en conclut que les relations de "croisement" et de "parallélisme" sont symétriques. $\diamond$

Ceci nous prouve aussi que si $S \sharp T$, alors $T$ intersecte toutes les composantes pleines associées à $S$ :

**Proposition 2.25** *Si $S$ et $T$ sont deux séparateurs minimaux qui se croisent, alors $T$ intersecte chacune des composantes connexes pleines associées à $S$.*

Remarquons que si $S$ et $T$ sont parallèles, alors $T$ est contenu dans un bloc $(S, C)$ de $S$. La relation de parallélisme possède une sorte de "pseudo-transitivité" :

**Proposition 2.26** *Soient $S, T$ et $U$ trois séparateurs minimaux de $G$. Si $T \| S$, $S \| U$ et $T$ et $U$ se trouvent dans des blocs différents de $S$, alors $T \| U$ comme le montre la figure 2.7.*

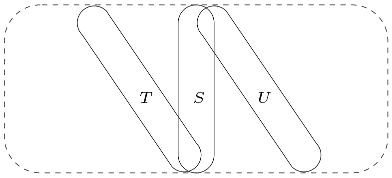


FIG. 2.7: "Transitivité" de la relation de parallélisme

On dit dans ce cas que $S$ *sépare* $T$ et $U$. Notons aussi que si $S \subset T$, alors $S$ et $T$ sont parallèles.

Utilisant le fait qu'un séparateur ne peut pas séparer deux sommets adjacents et le théorème 2.17, on déduit facilement :

**Lemme 2.27** *Soient $G$ un graphe, $S$ un séparateur minimal et $\Omega$ une clique de $G$. Alors $\Omega$ est incluse dans un bloc de $S$. Si $T$ est un séparateur minimal et $G[T]$ est une clique, alors $S \| T$. En particulier, les séparateurs minimaux d'un graphe triangulé sont deux à deux parallèles.*

Dans la suite, $G = (V, E)$ sera un graphe quelconque et $H = (V, F)$ sera une triangulation minimale de $G$. Nous avons remarqué que les séparateurs minimaux d'un graphe triangulé sont des cliques, par le théorème 2.17. S'impose alors l'idée naturelle que les triangulations minimales d'un graphe $G$ sont obtenues en considérant certaines familles de séparateurs de $G$ et les complétant pour les transformer en cliques, c'est-à-dire en rajoutant toutes les arêtes manquantes à l'intérieur de chaque séparateur considéré, comme dans le cas illustré par la figure 2.8.

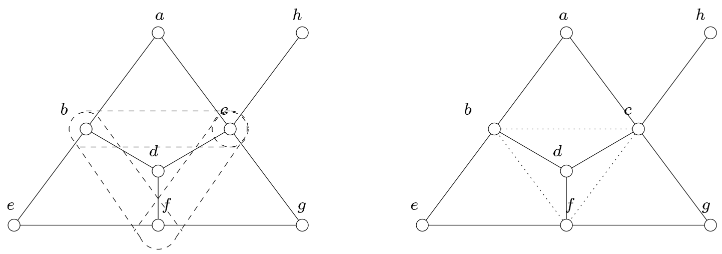


FIG. 2.8: Triangulations minimales et séparateurs minimaux

Si $\Gamma = \{S_1, \ldots, S_p\}$ est une famille de séparateurs minimaux de $G$, on note $G_\Gamma = (V, E \cup E')$, où $E' = \{xy | \exists S \in \Gamma \text{ t.q. } x, y \in S\}$ le graphe obtenu en complétant tous les séparateurs de $\Gamma$. La figure 2.8 montre le graphe $G_\Gamma$ pour

$\Gamma = \{\{b,c\},\{c\},\{b,f\},\{c,f\}\}$. Lorsque $\Gamma$ ne contient qu'un seul séparateur $S$ on note $G_S$ à la place de $G_{\{S\}}$.

Historiquement, la première étape a été d'introduire une notion de triangulation telle que tous les séparateurs de la triangulation soient aussi des séparateurs du graphe initial et qu'ils induisent les mêmes composantes connexes dans les deux graphes. Ces triangulations ne sont pas nécessairement minimales, comme on peut se convaincre en prenant comme triangulation d'un graphe quelconque son sur-graphe complet. Par contre, les triangulations minimales satisfont ces propriétés, fait prouvé par Kloks et Kratsch et Müller, par exemple dans [KKM95] :

**Lemme 2.28** *Soit $H$ une triangulation minimale d'un graphe $G$. Tout séparateur minimal $S$ de $H$ est aussi un séparateur minimal de $G$ et les composantes connexes, respectivement les composantes connexes pleines induites par $S$ dans $H$ sont les mêmes que les composantes connexes, respectivement les composantes connexes pleines induites par $S$ dans $G$ : $\mathcal{C}_H(S) = \mathcal{C}_G(S)$ et $\mathcal{C}^*_H(S) = \mathcal{C}^*_G(S)$.*

*De plus, $H = G_{\Delta_H}$, c'est-à-dire $H$ est obtenu en complétant dans $G$ les séparateurs de $\Delta_H$.*

*Preuve.* Soit $S$ un séparateur minimal de $H$. Puisque $G$ est un sous-graphe de $H$, $G \backslash S$ a aussi plusieurs composantes connexes, chaque composante connexe de $H \backslash S$ étant une réunion de composantes connexes de $G$. Montrons que les composantes connexes, respectivement les composantes connexes pleines induites par $S$ dans $G$ sont les mêmes que celles induites dans $H$.

Si ce n'est pas le cas, prouvons qu'il existe deux sommets $x$ et $y$ adjacents dans $H$, mais séparés dans $G$ par un séparateur minimal $S' \subseteq S$. En effet, si $\mathcal{C}_G(S) \neq \mathcal{C}_H(S)$, il existe deux composantes $C$ et $D$ de $G \backslash S$ qui sont reliées dans $H$ par une arête $xy$ avec $x \in C$ et $y \in D$. $S$ est un $x,y$-séparateur dans $G$, donc il existe un $x,y$-séparateur minimal $S'$ de $G$ avec $S' \subseteq S$. Si $\mathcal{C}_G(S) \neq \mathcal{C}_H(S)$ mais $\mathcal{C}^*_G(S) \neq \mathcal{C}^*_H(S)$, il existe une composante pleine $C$ induite par $S$ dans $H$ qui n'est pas pleine par rapport à $S$ dans $G$. Soit $S''$ l'ensemble des sommets de $S$ ayant, dans le graphe $G$, un voisin dans $C$. Comme $C \notin \mathcal{C}^*_G(S)$, il existe un sommet $x \in S \backslash S''$. Comme $C$ est pleine par rapport à $S$ dans $H$, le sommet $x$ est adjacent dans $H$ à un $y \in C$. Clairement $S'' \subset S$ sépare dans $G$ les sommets $x$ et $y$, en particulier il existe un séparateur minimal $S' \subset S$ de $G$ séparant $x$ et $y$ dans $G$.

Soit donc $S' \subseteq S$ un $x,y$-séparateur minimal de $G$, avec $xy \in E(H)$. Soit $H'$ le graphe obtenu à partir de $H$ en enlevant toutes les arêtes qui relient (dans $H$) les sommets de $C^x_G(S')$ à ceux de $C^y_G(S')$. Clairement $G \subseteq H' \subset H$, la deuxième inclusion est stricte car on a au moins supprimé l'arête $xy$. Par le lemme 2.8, $H'$ n'est pas un graphe triangulé, il contient donc un cycle $\mu$, sans corde. Ce cycle doit forcément contenir des sommets dans les composantes $C^x_G(S')$ et $C^y_G(S')$, sinon ce serait aussi un cycle induit de $H$. Comme $S'$ sépare dans $H'$ les composantes $C^x_G(S')$ et $C^y_G(S')$, notre cycle intersecte $S'$ en au moins deux sommets $c$ et $d$. Mais $H'[S']$ étant une clique, $cd$ est donc une corde du cycle $\mu$, d'où la contradiction. Ceci prouve que $S$ induit les mêmes composantes connexes et les mêmes composantes connexes pleines dans $G$ et dans $H$.

Montrons maintenant que $H = G_{\Delta_H}$. Par le théorème de Dirac, tous les séparateurs minimaux de $H$ sont des cliques, d'où $G_{\Delta_H} \subseteq H$. Supposons qu'il existe une arête $ab$ dans $E(H) \backslash E(G_{\Delta_H})$ et soit $H' = (V(H), E(H) \backslash \{ab\})$. Alors $H'$ n'est pas un graphe triangulé, donc il contient un cycle sans corde. Ce cycle doit passer par $a$ et $b$ et de plus il a au plus quatre sommets, sinon il ne pourrait pas être triangulé par simple ajout d'une arête. On note $[a, c, b, d]$ ce cycle. Il existe alors un $c, d$-séparateur minimal de $H'$ qui contient $a$ et $b$. Ce sera aussi un séparateur minimal de $H$, contredisant le fait que $ab \notin G_{\Delta_H}$. ◇

Nous venons de vérifier que les triangulations minimales de $G$ sont effectivement obtenues en complétant certains séparateurs de $G$. Il s'agit maintenant de caractériser les familles de séparateurs minimaux à compléter pour obtenir des triangulations minimales. Ce résultat apparaît dans les travaux de Parra et Scheffler [PS97].

Nous avons remarqué dans le lemme 2.27 que les séparateurs minimaux d'un graphe triangulé sont deux à deux parallèles. Soient $S$ et $T$ deux séparateurs minimaux d'une triangulation minimale $H$ de $G$. Puisque $S \| T$, $T$ sera contenu dans un bloc $(S, C)$ de $S$ dans $H$. Mais les composantes connexes et en conséquence les blocs de $S$ dans $G$ sont les mêmes que dans $H$, par le lemme précédent. On en déduit que $S$ et $T$ sont parallèles dans $G$. Donc les séparateurs minimaux de $H$ sont deux à deux parallèles dans $G$. Avant d'aller plus loin, regardons ce qui se passe lorsqu'on complète une famille de séparateurs minimaux deux à deux parallèles d'un graphe.

**Lemme 2.29** *Soit $\Gamma \subseteq \Delta_G$ une famille de séparateurs minimaux deux à deux parallèles et soit $T \in \Delta_G$ un $a, b$-séparateur minimal parallèle à tout $S \in \Gamma$. Alors $T$ est un séparateur minimal pour tout graphe $H$ avec $G \subseteq H \subseteq G_\Gamma$ et les composantes connexes induites par $T$ dans $H$ sont les mêmes que celles induites dans $G$.*

*Preuve.* Tout $S \in \Gamma$ étant parallèle à $T$, $S$ intersecte au plus une composante connexe de $G \backslash T$. Il s'ensuit que toute composante connexe de $G \backslash T$ restera une composante connexe dans $G_\Gamma \backslash T$, sinon au moins deux composantes de $G \backslash T$ seraient fusionnées dans $G_\Gamma \backslash T$ ce qui signifierait l'existence d'un séparateur $S \in \Gamma$ croisant $T$. Il en résulte que pour tout graphe $H$ pris en sandwich entre $G$ et $G_\Gamma$ les composantes connexes de $H \backslash T$ sont les mêmes que celles de $G \backslash T$ et de $G_\Gamma \backslash T$. En particulier $C^a_G(T)$ et $C^b_G(T)$ seront des composantes connexes de $H \backslash T$, pleines par rapport à $T$, donc $T$ est aussi un $a, b$-séparateur minimal de $H$. ◇

On voit donc très facilement que les séparateurs de la triangulation minimale $H$ forment une famille maximale de séparateurs deux à deux parallèles de $G$.

**Proposition 2.30** *Soit $H$ une triangulation minimale d'un graphe $G$. Les séparateurs minimaux de $H$ forment une famille maximale (par inclusion) de séparateurs parallèles de $G$ et $H = G_{\Delta_H}$.*

*Preuve.* On sait par le lemme 2.29 que $H = G_{\Delta_H}$.

Nous avons remarqué plus haut que les séparateurs minimaux de $H$ sont deux à deux parallèles dans $G$. Soit maintenant $T \in \Delta_G$ un séparateur minimal parallèle à tout $S \in \Delta_H$. Par le lemme 2.29, $T$ est aussi un séparateur minimal de $H$. Donc $T \in \Delta_H$, ce qui achève la démonstration. ◇

Nous allons prouver que, réciproquement, en complétant une famille maximale de séparateurs minimaux de $G$ nous allons obtenir une triangulation minimale du graphe. Avant cela, remarquons que si l'on complète une famille (pas nécessairement maximale) de séparateurs parallèles de $G$, les séparateurs minimaux du graphe ainsi obtenu sont aussi des séparateurs minimaux de $G$.

**Lemme 2.31** *Soit $\Gamma \subseteq \Delta_G$ une famille de séparateurs minimaux deux à deux parallèles. Notons $H = G_\Gamma$. Alors tout $a, b$-séparateur minimal $T$ de $H$ est aussi un $a, b$-séparateur minimal de $G$ et pour tout $S \in \Gamma$, $T$ et $S$ sont parallèles dans $G$. De plus, les composantes connexes et les composantes connexes pleines associées à $T$ dans $G$ sont les mêmes que dans $H$.*

*Preuve.* Soit $T$ un séparateur minimal de $H$. Montrons que $T$ induit les mêmes composantes connexes et les mêmes composantes pleines dans $G$ et dans $H$. Si ce n'est pas le cas, comme dans la preuve du lemme 2.28 nous prouvons qu'il existe deux sommets $x$ et $y$ adjacents dans $H$ mais séparés dans $G$ par un séparateur minimal $T' \subseteq T$. Soit $C$ une composante connexe de $G\backslash T$. Si $C$ n'est pas une composante connexe de $H\backslash T$, on en déduit qu'il existe une arête $xy$ de $E(H)\backslash E(G)$ dont les extrémités se trouvent dans des composantes connexes différentes, $C_G^x$ et respectivement $C_G^y$, de $G\backslash T$. Il existe alors un $x, y$-séparateur minimal $T'$ dans $G$, avec $T' \subseteq T$. Si $C$ une composante connexe de $G\backslash T$ non-pleine par rapport à $T$ dans $G$, mais qui est pleine dans $H$, on pose $T''$ l'ensemble des sommets de $C$ ayant, dans le graphe $G$, au moins un voisin dans $T$. Comme $C$ est pleine par rapport à $T$ dans $H$ mais pas dans $G$, on a dans $H$ une arête $xy$ avec $x \in T\backslash T''$ et $y \in C$. Clairement $T''$ sépare $x$ et $y$ dans $G$, donc il existe dans $G$ un $x, y$-séparateur minimal $T' \subset T$.

Nous avons donc un $x, y$-séparateur minimal $T'$ dans $G$ tel que $T' \subseteq T$ et $xy \in E(H)$. Comme $xy \in E(H)\backslash E(G)$, il existe un séparateur minimal $S \in \Gamma$ contenant $x$ et $y$. Clairement $T' \sharp S$ dans le graphe $G$. Mais les composantes connexes induites par $S$ dans $H = G_\Gamma$ sont les mêmes que dans $G$, car $S$ est parallèle dans $H$ à tous les éléments de $\Gamma$ et donc toute arête rajoutée dans $H$ a ses deux extrémités dans un même bloc de $S$. En conséquence, $T'$ intersecte des composantes connexes différentes de $H\backslash S$. Il s'ensuit que $T \supseteq T'$ intersecte lui aussi deux composantes connexes de $H\backslash S$, donc $T \sharp S$ dans $H$. Comme $S$ est une clique dans $H$, ceci contredit le lemme 2.27.

Nous avons prouvé que $\mathcal{C}_G(T) = \mathcal{C}_H(T)$ et $\mathcal{C}_G^*(T) = \mathcal{C}_H^*(T)$. Comme les composantes connexes pleines associées à $T$ sont les mêmes dans les deux graphes, par la proposition 2.14, $T$ sera un $a, b$-séparateur minimal de $G$. $\diamond$

On en déduit :

**Proposition 2.32** *Soit $G$ un graphe quelconque et soit $\Gamma \subseteq \Delta_G$ une famille de séparateurs deux à deux parallèles, maximale par inclusion. Alors $H = G_\Gamma$ est une triangulation minimale de $G$ et $\Delta_H = \Gamma$.*

*Preuve.* Soit $T$ un séparateur minimal quelconque de $H$. Nous allons montrer que $T$ est une clique, ce qui, par application du théorème de Dirac (théorème 2.17) entraînera que $H$ est un graphe triangulé, c'est-à-dire une triangulation de $G$. Par le lemme 2.31, $T$ est aussi un séparateur minimal de $G$, parallèle à tout $S \in \Gamma$. Par maximalité de la famille $\Gamma$ on a $T \in \Gamma$, donc $T$ est une clique dans $H$.

Montrons que $H$ est une triangulation minimale. Supposons qu'il existe une triangulation $H'$ de $G$ avec $H' \subseteq H$ et montrons que $H' = H$. Soit $xy$ une arête de $E(H)\backslash E(G)$, il existe un séparateur minimal $T \in \Gamma$ qui contient les deux sommets $x$ et $y$ par définition de $G_\Gamma$. Ce séparateur est aussi un séparateur minimal de $G$ et par le lemme 2.29 c'est aussi un séparateur minimal de $H'$. Comme $H'$ est triangulé, $T$ induit une clique dans $H'$, ainsi $xy \in E(H')$ d'où $H = H'$. $\diamond$

Les propositions 2.30 et 2.32 sont résumées par Parra et Scheffler [Par96], [PS97] dans le théorème qui suit :

**Théorème 2.33 ([PS97])**

*Soit $H$ une triangulation minimale d'un graphe $G$. On a que $\Delta_H$ est un ensemble maximal de séparateurs deux à deux parallèles de $G$ et $H = G_{\Delta_H}$.*

*Soit $\Gamma \in \Delta_G$ un ensemble maximal de séparateurs deux à deux parallèles de $G$. Alors $H = G_\Gamma$ est une triangulation minimale de $G$ et $\Delta_H = \Gamma$.*

Soulignons encore une fois que les séparateurs minimaux d'une triangulation minimale $H$ de $G$ se comportent de façon homogène dans les deux graphes, c'est-à-dire qu'ils induisent les mêmes composantes connexes et les mêmes composantes connexes pleines dans $G$ et dans $H$.

Nous savons maintenant comment sont formées les triangulations minimales d'un graphe $G$, à partir de ses séparateurs minimaux. Elles s'obtiennent toujours en choisissant une famille maximale de séparateurs minimaux deux à deux parallèles de $G$ et en les transformant en cliques. Donc pour trouver une triangulation minimale, nous devons regarder *globalement* les séparateurs minimaux de $G$ et la relation de parallélisme entre eux. Dans sa thèse, Parra conclut que si l'on sait calculer en temps polynomial pour une classe de graphes *toutes* les familles maximales de séparateurs deux à deux parallèles, la largeur arborescente et la complétion minimale de ces graphes sont calculables en temps polynomial.

Toutefois le nombre de familles de ce type est exponentiel même pour des classes de graphes très simples, comme les cycles. Prenons en effet comme exemple un cycle de longueur de longueur $2p$, noté $C_{2p} = [0, 1, \ldots, 2p-1]$. Les séparateurs minimaux de ce cycle sont les ensembles $\{i, j\}$, avec $1 \leq i, j \leq 2p$ et $|i-j| > 1$, donc $C_{2p}$ a moins de $(2p)^2$ séparateurs minimaux. Remarquons sur la figure 2.9 que deux séparateurs $\{i, j\}$ et $\{i', j'\}$ se croisent si et seulement si les segments $[ij]$ et $[i'j']$ se croisent, ce qui nous permet de visualiser facilement les familles de séparateurs parallèles. Considérons maintenant les séparateurs $S_i = \{i, 2p-i\}$, pour $1 \leq i \leq p-1$, et les séparateurs $T_i = \{i, 2p-i-1\}$ et $T'_i = \{i+1, 2p-i\}$ pour $1 \leq i \leq p-2$. Il est facile de voir que pour tout choix de $U_i \in \{T_i, T'_i\}$, $1 \leq i \leq p-2$, la famille de séparateurs $\{S_1, U_1, S_2, U_2, \ldots, S_{p-2}, U_{p-2}, S_{p-1}\}$ est une famille maximale de séparateurs parallèles de $C_{2p}$. On voit donc que $C_{2p}$ a plus de $2^{p-2}$ familles maximales de séparateurs deux à deux parallèles.

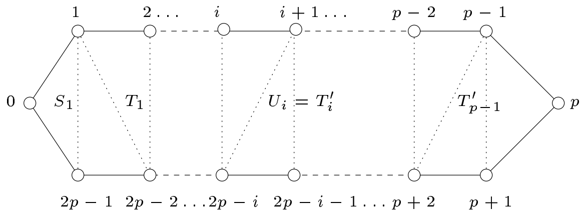


FIG. 2.9: Séparateurs du cycle $C_{2p}$

La caractérisation des triangulations minimales donnée par le théorème précédent nécessite une vue globale sur la structure des séparateurs minimaux. Même pour les graphes avec un nombre polynomial de séparateurs minimaux, nous ne pouvons pas calculer facilement toutes les familles maximales de séparateurs parallèles. Par la suite, nous tenterons de regarder *localement* les séparateurs minimaux d'un graphe. Nous cherchons un objet qui caractérise toutes les triangulations minimales d'un graphe et qui soit calculable en temps polynomial.

# Chapitre 3

# Triangulations minimales

Dans ce chapitre nous donnerons une nouvelle caractérisation des triangulations minimales d'un graphe, d'un point de vue plus local. Nous introduirons notamment les notions de *clique maximale potentielle* et de *famille maximale de séparateurs voisins*. Ces objets caractérisent toutes les cliques maximales de toutes les triangulations minimales d'un graphe $G$.

Les notions de clique maximale potentielle et de famille maximale de séparateurs voisins sont complètement nouvelles. Nous les avons introduites dans [BT98a, BT98b] et elles nous ont permis d'unifier les algorithmes déjà existants de calcul de la largeur arborescente et de la complétion minimales pour certaines classes de graphes avec un nombre polynomial de séparateurs. Plus exactement, nous avons montré que tous ces algorithmes calculent de manière implicite toutes les cliques maximales potentielles du graphe en entrée. De plus, nous avons utilisé ces notions pour résoudre ces problèmes pour une autre classe, à savoir les graphes faiblement triangulés. Tout ceci nous donne de bonnes raisons de penser que le calcul de ces paramètres de triangulation pour tous les graphes avec "peu" de séparateurs minimaux passe par la connaissance des cliques maximales potentielles. En tout cas nous montrerons dans les chapitres suivants que ces objets sont suffisants pour calculer la largeur arborescente et la complétion minimale d'un graphe.

Nous ne donnerons dans ce chapitre que les définitions et les propriétés structurelles fondamentales des cliques maximales potentielles et des familles maximales de séparateurs voisins. Les chapitres suivants sont entièrement basés sur ces notions et illustrent leurs applications algorithmiques.

Pour avoir un aperçu sur les cliques maximales potentielles, reprenons l'exemple du cycle. Toute triangulation est obtenue en complétant une famille de séparateurs deux à deux parallèles, comme sur la figure 3.1. Nous obtenons des "régions" triangulaires entre ces séparateurs et les cliques maximales des triangulations sont toutes des triplets de sommets $\{i, j, k\}$. Cela nous suggère que toutes les triangulations du cycle sont déterminées par ces "cliques maximales potentielles" formées par des triplets de sommets.

Nous nous donnerons ici les moyens de reconnaître une clique maximale poten-

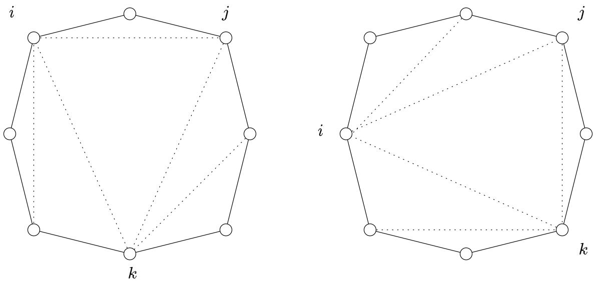


FIG. 3.1: Triangulations du cycle

tielle : nous répondrons à la question “étant donné un ensemble de sommets $K$ d'un graphe $G$, existe-t-il une triangulation minimale $H$ de $G$ telle que $K$ soit une clique maximale de $H$ ?”.

Il faudra ensuite attendre les résultats des chapitres suivants pour se convaincre que ces cliques maximales potentielles suffisent pour calculer la largeur arborescente et la complétion minimale d'un graphe et enfin que, pour de larges classes de graphes, ces objets sont calculables en temps polynomial.

## 3.1 Cliques maximales des graphes triangulés

Nous commençons par quelques observations sur la formation des cliques maximales des graphes triangulés en fonction des séparateurs minimaux du graphe, dans le but de les étendre aux cliques des triangulations minimales des graphes quelconques.

Introduisons d'abord une nouvelle notion. En considérant une famille de séparateurs deux à deux parallèles et telle qu'aucun d'entre eux ne sépare deux autres - les séparateurs $S_1, S_2, S_3$ de la figure 3.2 illustrent cette situation - on voit qu'ils déterminent une zone du graphe qui se trouve “entre” eux. Plus formellement :

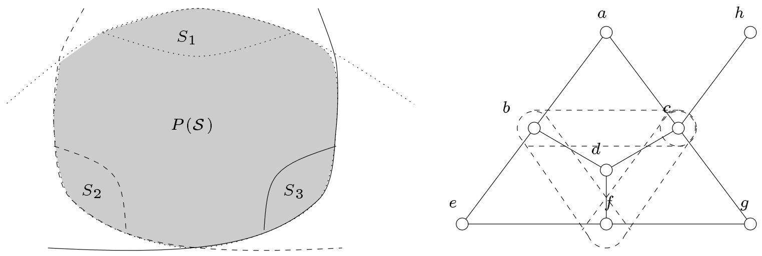


FIG. 3.2: Partie entre une famille de séparateurs

**Définition 3.1** *Soit $G$ un graphe quelconque et $\mathcal{S} \subseteq G$ une famille de séparateurs minimaux telle que pour tout $S \in \mathcal{S}$, il existe un bloc $(S, C_G(S))$ contenant tous les séparateurs de $\mathcal{S}$. Si $\mathcal{S}$ n'a pas d'élément maximum pour l'inclusion, on définit la* partie entre *les séparateurs de $\mathcal{S}$ par*

$$P(\mathcal{S}) = \bigcap_{S \in \mathcal{S}} (S, C_G(S))$$

Sur l'exemple de la figure 3.2, si l'on prend les séparateurs entourés en pointillés $\mathcal{S} = \{\{b,c\},\{c\},\{b,f\},\{c,f\}\}$ on obtient $P(\mathcal{S}) = \{b,c,d,f\}$.

Si la partie entre une famille $\mathcal{S}$ de séparateurs minimaux existe, les séparateurs de $\mathcal{S}$ doivent être deux à deux parallèles. En effet, si $S,T \in \mathcal{S}$, puisque $T \subseteq (S, C_G(S))$ on a $S \| T$. D'un autre côté, si l'on se donne une famille $\mathcal{S}'$ de séparateurs minimaux deux à deux parallèles, sans élément maximum par inclusion, la partie entre ces séparateurs existe si et seulement si aucun $S \in \mathcal{S}'$ ne sépare deux éléments $T, U \in \mathcal{S}'$. Notons aussi que, d'après la définition, si $P(\mathcal{S})$ existe, alors elle contient tous les séparateurs de $\mathcal{S}$.

Dans la suite nous nous intéresserons aux cliques maximales d'un graphe triangulé et à leur position par rapport aux séparateurs qu'elles contiennent. Si $K$ est un ensemble de sommets de $G$, nous noterons par $\Delta_G(K)$ l'ensemble des séparateurs minimaux de $G$ contenus dans $K$.

Commençons par une observation très facile :

**Lemme 3.2** *Soient $H$ un graphe triangulé et $\Omega$ une de ses cliques maximales. Alors soit $\Delta_H(\Omega)$ a un élément maximum par inclusion, soit $P(\Delta_H(\Omega))$ existe et contient $\Omega$.*

*Preuve.* Par le lemme 2.27, pour tout $S \in \Delta_H(\Omega)$, la clique $\Omega$ est contenue dans un bloc $(S, C_H(S))$, donc tous les éléments de $\Delta_H(\Omega)$ sont inclus dans ce bloc. Si $\Delta_H(\Omega)$ n'a pas d'élément maximum, par définition de la partie entre nous avons $\Omega \subseteq P(\Delta_H(\Omega))$. ◇

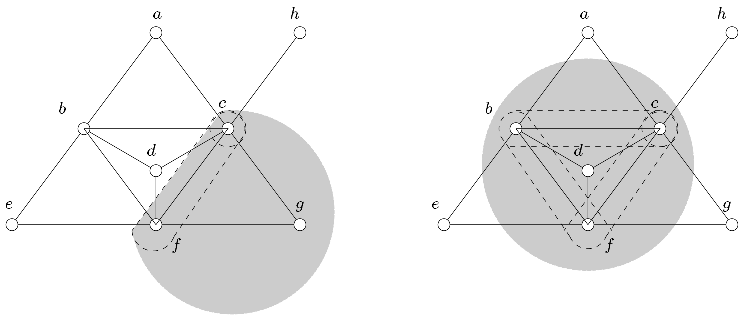


FIG. 3.3: Cliques maximales

Nous avons maintenant tous les ingrédients pour établir une relation entre une clique maximale $\Omega$ d'un graphe triangulé $H$ et l'ensemble des séparateurs minimaux qu'elle contient.

**Théorème 3.3** *Soit $H$ un graphe triangulé et $\Omega$ une clique maximale de $H$. Si tous les éléments de $\Delta_H(\Omega)$ sont contenus dans un séparateur minimal $S \in \Delta_H(\Omega)$, alors $\Omega$ est un bloc $(S, C)$ de $S$. Si $\Delta_G(\Omega)$ n'a pas d'élément maximum par inclusion, alors $\Omega = P(\Delta_H(\Omega))$.*

*Preuve.* Supposons d'abord que $\bigcup_{T \in \Delta_H(\Omega)} T = S$, avec $S \in \Delta_H(\Omega)$. Par le lemme 2.27, $\Omega$ est contenue dans un bloc $(S, C)$ de $S$. Supposons qu'il existe un sommet $x \in C$ qui ne soit pas dans $\Omega$. Considérons alors dans un arbre de cliques de $H$ une chaîne $\Omega, \Omega_1, \ldots, \Omega_p$ de cliques adjacentes de sorte que $x \in \Omega_p$. Alors $T = \Omega \cap \Omega_1$ est un séparateur minimal de $H$ contenu dans $\Omega$, donc un élément de $\Delta_H(\Omega)$. Puisque $x \notin \Omega$, par le lemme 2.21 on a que $T$ sépare $x$ et tout sommet de $\Omega \backslash T$. Comme $T \subseteq S$, nous obtenons aussi que $S$ sépare $x$ et $\Omega \backslash S$, contredisant le fait que $x$ et $\Omega$ sont dans le même bloc de $S$. On en conclut que $\Omega = (S, C)$.

Supposons maintenant qu'aucun élément de $\Delta_H(\Omega)$ ne contienne tous les autres. D'après le lemme 3.2, $\Omega \subseteq P(\Delta_H(\Omega))$. Supposons qu'il existe un $x \in P(\Delta_H(\Omega))$ qui ne soit pas contenu dans $\Omega$. Comme pour le premier cas, prenons dans un arbre de cliques de $H$ une chaîne $\Omega, \Omega_1, \ldots, \Omega_p$ de cliques adjacentes telle que $x \in \Omega_p$. Alors $T = \Omega \cap \Omega_1$ est un séparateur minimal appartenant à $\Delta_H(\Omega)$ et qui sépare $x$ de tout élément de $\Omega \backslash T$, d'après le lemme 2.21. Soit $S$ un autre séparateur minimal de $\Delta_H(\Omega)$, non contenu dans $T$. Remarquons que $S$ existe parce que $T$ n'est pas maximum par inclusion dans $\Delta_H(\Omega)$. Soit $y$ un sommet de $S \backslash T$. Nous savons donc que $T$ sépare $x$ et $y$. Mais ceci contredit le fait que, puisque $x, y \in P(\Delta_H(\Omega))$, les sommets $x$ et $y$ doivent se trouver dans un même bloc de $T$. Il s'ensuit que $\Omega = P(\Delta_H(\Omega))$. ◇

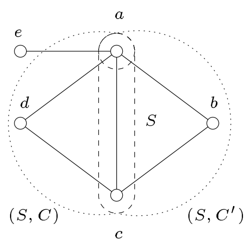


FIG. 3.4: Cliques maximales - cas dégénéré

Nous allons dire par la suite que les éléments de $\Delta_H(\Omega)$ *bordent* la clique maximale $\Omega$. En effet, d'une façon ou d'une autre, ils séparent $\Omega$ du reste du graphe. Notons ensuite que si la famille de séparateurs $\Delta_H(\Omega)$ n'a pas d'élément maximum par inclusion, elle ne peut border que la clique maximale $\Omega = P(\Delta_H(\Omega))$, car $P(\Delta_H(\Omega))$ est définie de façon unique. Par contre, deux cliques maximales différentes $\Omega$ et $\Omega'$ peuvent être bordées par la même famille de séparateurs, comme le montre la figure 3.4. Ceci peut arriver uniquement si la famille de séparateurs $\Delta_H(\Omega) = \Delta_H(\Omega')$ a un élément $S$ maximum par inclusion, car on peut avoir plusieurs blocs de $S$ formant des cliques maximales. Dans notre exemple, $\Omega = (S, C)$ et $\Omega' = (S, C')$. C'est pourquoi nous n'avons pas défini $P(\Delta_H(\Omega))$ dans ce cas "dégénéré".

## 3.2 Cliques maximales potentielles

Dans cette section nous montrerons comment se forment les cliques maximales des triangulations minimales d'un graphe $G$ quelconque. Nous avions déjà constaté que, si $H$ est une triangulation minimale d'un graphe $G$, les séparateurs minimaux de $H$ sont non seulement des séparateurs minimaux de $G$, mais ils se comportent en quelque sorte "de la même façon" dans les deux graphes. En partant de cette idée il est naturel de penser que l'on pourrait caractériser les cliques maximales de $H$ à partir des séparateurs de $G$, d'une manière très similaire au théorème 3.3.

Rappelons que la question à laquelle nous voulons répondre est : étant donné un ensemble $K$ de sommets de $G$, comment savoir s'il existe une triangulation minimale $H$ de $G$ dans laquelle $K$ est une clique maximale ?

**Définition 3.4** *Un ensemble de sommets $\Omega$ d'un graphe $G$ est appelé* clique maximale potentielle *s'il existe une triangulation minimale $H$ de $G$ telle que $\Omega$ soit une clique maximale de $H$.*

Signalons encore une fois que la terminologie de "clique maximale potentielle" est reprise d'après les travaux indépendants de Kloks, Kratsch et Müller [KKM98].

Cette définition est de nature combinatoire, au sens où pour identifier une clique maximale potentielle de $G$ il nous faut passer par les triangulations minimales du graphe. L'objectif de cette section est de montrer que les cliques maximales potentielles peuvent se définir de façon parfaitement locale - algorithmique, si l'on veut - en restant dans le graphe $G$ et sans utiliser les triangulations minimales.

La clique maximale potentielle est fortement liée aux séparateurs qu'elle contient, tout comme une clique maximale d'un graphe triangulé.

**Définition 3.5** *Une famille $\mathcal{S}$ de séparateurs minimaux d'un graphe $G$ est une* famille maximale de séparateurs voisins *s'il existe une clique maximale potentielle $\Omega$ de $G$ telle que $\mathcal{S} = \Delta_G(\Omega)$.*

Nous justifierons plus loin l'appellation de "séparateurs voisins".

Avant de donner une généralisation du théorème 3.3, qui nous permettra de reconnaître les cliques maximales potentielles et les familles maximales de séparateurs voisins, nous avons besoin de quelques résultats.

**Proposition 3.6** *Soit $H$ une triangulation minimale d'un graphe $G$ et soit $T$ un séparateur minimal de $G$ tel que $H[T]$ soit une clique. Alors $T$ est aussi un séparateur minimal de $H$.*

*Preuve.* Nous savons que $\Delta_H$ est une famille maximale de séparateurs deux à deux parallèles de $G$, par le théorème 2.33. Soit $S$ un séparateur minimal quelconque de $H$. Par le lemme 2.27, la clique $T$ est contenue dans un certain bloc $(S, C)$ de $S$ dans $H$. Mais $(S, C)$ est aussi un bloc de $S$ dans $G$, donc $S$ et $T$ sont parallèles dans $G$. On a donc que $T$ est parallèle dans $G$ à tout séparateur minimal de $\Delta_H$. Comme $\Delta_H$ est une famille maximale par inclusion de séparateurs parallèles de $G$ on en déduit que $T \in \Delta_H$. $\diamond$

**Corollaire 3.7** *En particulier, la proposition 3.6 prouve que si $T$ est un séparateur minimal de $G$ contenu dans un séparateur minimal $S$ tel que $S \in \Delta_H$, alors $T \in \Delta_H$.*

Mais nous allons utiliser ce résultat surtout sous la forme suivante, qui nous dit qu'une clique d'une triangulation minimale $H$ de $G$ contient les mêmes séparateurs dans $G$ et dans $H$ :

**Proposition 3.8** *Soient $G$ un graphe et $H$ une triangulation minimale de $G$. Soit $\Omega$ une clique de $H$. Alors $\Delta_H(\Omega) = \Delta_G(\Omega)$.*

*Preuve.* Comme tout séparateur minimal de $H$ est aussi séparateur minimal de $G$, on a clairement $\Delta_H(\Omega) \subseteq \Delta_G(\Omega)$.

Soit maintenant $S$ un séparateur de $G$ contenu dans $\Omega$. Comme $H[S]$ est une clique, par la proposition 3.6 $S$ est un séparateur minimal de $H$ donc $S \in \Delta_H(\Omega)$. $\diamond$

Le lemme suivant nous sert à montrer que si $\Omega$ est une clique maximale d'une triangulation $H = G_\Gamma$ de $G$, alors $\Omega$ est aussi une clique dans le graphe $G_{\Delta_G(\Omega)}$ obtenu à partir de $G$ en complétant uniquement les séparateurs minimaux de $G$ contenus dans $\Omega$. Ceci nous permettra de définir localement les cliques potentielles, sans passer par une triangulation minimale du graphe.

**Lemme 3.9** *Soient $H$ un graphe triangulé, $\Omega$ une clique maximale et $S$ un séparateur minimal de $H$ qui intersecte $\Omega$ ($S$ n'est pas forcément contenu dans $\Omega$). Il existe alors un séparateur minimal $T \subset \Omega$ de $H$ tel que $S \cap \Omega \subseteq T$.*

*Preuve.* Si $S \subseteq \Omega$ la propriété est triviale.

Sinon, considérons un arbre de cliques $\mathcal{T}$ du graphe $H$. Par le théorème 2.20, il existe deux cliques maximales $\Omega_1$ et $\Omega_2$ adjacentes dans $\mathcal{T}$ telles que $S = \Omega_1 \cap \Omega_2$. Les deux cliques sont différentes de $\Omega$. Comme $\mathcal{T}$ est un arbre et que $\Omega_1$, $\Omega_2$ sont des nœuds adjacents, l'un des deux nœuds, mettons $\Omega_1$, est plus proche de $\Omega$ que $\Omega_2$. De plus, l'unique chaîne de $\Omega$ à $\Omega_2$ dans l'arbre $\mathcal{T}$ passe par $\Omega_1$. Soit $\Omega'$ la clique adjacente à $\Omega$ sur cette chaîne. Nous avons donc dans $\mathcal{T}$ la chaîne $[\Omega, \Omega', \ldots, \Omega_1, \Omega_2]$, avec éventuellement $\Omega' = \Omega_1$.

Par la propriété d'intersection des cliques, $\Omega'$ contient à la fois les ensembles $\Omega \cap \Omega_1$ et $\Omega \cap \Omega_2$. Puisque $S = \Omega_1 \cap \Omega_2$ on en déduit que $S \cap \Omega$ est contenu dans $\Omega \cap \Omega'$, qui est le séparateur minimal $T$ voulu. $\diamond$

On en déduit :

**Proposition 3.10** *Soit $H$ une triangulation minimale d'un graphe $G$. Soit $\Omega$ une clique maximale de $H$. Alors $\Omega$ est une clique dans le graphe $G_{\Delta_G(\Omega)}$.*

*Preuve.* Montrons que toute arête de $H[\Omega]$ est aussi une arête de $G_{\Delta_G(\Omega)}[\Omega]$. Toute arête de $H[\Omega]$ est soit une arête de $G$, ou bien ses extrémités $x$ et $y$ sont sur un même séparateur $S$ de $H$. Pour le deuxième cas, par le lemme 3.9, il existe un séparateur minimal $T \subseteq \Omega$ de $H$ qui contient $S \cap \Omega$. $T$ est aussi un séparateur minimal de $G$ car $\Delta_H \subseteq \Delta_G$. On en conclut que $xy$ est aussi une arête de $G_{\Delta_G(\Omega)}[\Omega]$, donc ce dernier graphe est une clique. $\diamond$

Nous pouvons donner maintenant une première caractérisation des cliques maximales potentielles d'un graphe, qui nous fournira un moyen algorithmique de reconnaître ces cliques. Pour plus de lisibilité, nous donnons deux théorèmes de caractérisation, correspondant aux deux cas rencontrés lors de la caractérisation des cliques maximales d'un graphe triangulé.

Regardons ce qui se passe si $H$ est une triangulation minimale de $G$ et $\Omega$ est une clique maximale de $H$. Nous considérons ici uniquement le cas du théorème 3.3 où $\Omega$ est un bloc $(S, C)$ de $H$, c'est-à-dire que $S$ est maximum par inclusion dans $\Delta_H(\Omega)$. Comme les blocs de $S$ sont les mêmes dans $G$ que dans $H$, $\Omega$ est aussi le bloc $(S, C)$ de $G$. D'après la proposition 3.10, $\Omega$ est une clique dans le graphe $G_{\Delta_G(\Omega)}$ obtenu à partir de $G$ en complétant localement les séparateurs minimaux de $G$ inclus dans $\Omega$.

Beaucoup plus intéressant, on peut donner de la même manière une condition nécessaire et suffisante pour qu'un ensemble de sommets soit une clique maximale potentielle, comme nous le montre le théorème suivant qui reprend en détail ces arguments.

**Théorème 3.11** *Soit $G$ un graphe et $\Omega$ un ensemble de sommets de $G$. Supposons que $\Delta_G(\Omega)$ ait un élément maximum $S$, i.e. chaque $T \in \Delta_G(\Omega)$ est contenu dans $S$. Alors $\Omega$ est une clique maximale potentielle si et seulement si les deux conditions sont satisfaites :*

1. *$\Omega$ est un bloc $(S, C)$ de $S$ dans $G$.*
2. *$G_{\Delta_G(\Omega)}[\Omega]$ est une clique.*

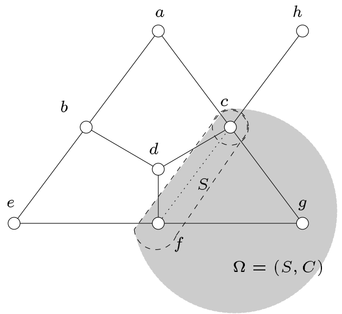


FIG. 3.5: Clique maximale potentielle : premier cas

*Preuve.*

"⇒" Supposons que $\Omega$ soit une clique maximale potentielle et considérons $H$ une triangulation minimale de $G$ dans laquelle $\Omega$ est une clique maximale. Notons que $\Delta_H(\Omega) = \Delta_G(\Omega)$, d'après la proposition 3.8. Il s'ensuit que $S$ est l'élément maximum de $\Delta_H(\Omega)$ et, d'après le théorème 3.3, il existe un bloc $(S, C)$ de $S$ dans $H$ tel que $\Omega = (S, C)$. Celui-ci est aussi un bloc de $S$ dans $G$, ce qui prouve la première assertion.

Selon la proposition 3.10, $G_{\Delta_G(\Omega)}[\Omega]$, qui est d'ailleurs le même que $G_S[\Omega]$, est une clique.

"$\Leftarrow$" Montrons maintenant que si $\Omega$ satisfait les deux conditions du théorème, il existe une triangulation minimale $H$ de $G$ telle que $\Omega$ soit une clique maximale de $H$. Tout d'abord, prouvons que les séparateurs minimaux de $\Delta_G(\Omega)$ sont deux à deux parallèles. En effet, il suffit de considérer une triangulation $H'$ de $G$ dont $S$ serait un séparateur minimal. On aura, par le corollaire 3.7, que tous les séparateurs minimaux $T \subset S$ de $G$ sont des séparateurs minimaux de $H'$, donc ils sont deux à deux parallèles dans $G$. On peut donc considérer une triangulation minimale $H$ de $G$ telle que $\Delta_G(\Omega) \subseteq \Delta_H$.

Comme $\Omega = (S, C)$ est une clique de $H$ par la deuxième condition, il nous faut encore montrer qu'elle est maximale. En effet, si $x$ est un sommet de $H$ adjacent à tous les sommets de $C$, comme $(S, C)$ est un bloc de $S$ dans $H$ la seule possibilité est $x \in S \cup C$. $\diamond$

Regardons maintenant ce qui se passe dans le cas non dégénéré. Considérons pour cela une triangulation minimale $H$ de $G$ et une clique maximale $\Omega = P(\Delta_H(\Omega))$ de $H$. Comme les séparateurs minimaux de $H$ induisent les mêmes composantes connexes dans $G$, la partie entre les séparateurs $\Delta_H(\Omega)$ est la même dans $G$ et dans $H$. Plus encore, d'après la proposition 3.8, $\Delta_H(\Omega) = \Delta_G(\Omega)$, donc $\Omega = P(\Delta_G(\Omega))$ dans $G$. Aussi, selon la proposition 3.10, $\Omega$ est une clique dans le graphe $G_{\Delta_G(\Omega)}$ obtenu en complétant dans $G$ tous les séparateurs de $\Delta_G(\Omega)$.

Le théorème qui suit reprend ces observations et établit une réciproque, comme pour le premier cas, en montrant comment reconnaître ce type de clique potentielle.

**Théorème 3.12** *Soit $G$ un graphe et $\Omega$ un ensemble de sommets de $G$. Supposons que $\Delta_G(\Omega)$ n'ait pas d'élément maximum pour l'inclusion. Alors $\Omega$ est une clique maximale potentielle si et seulement si les deux conditions sont satisfaites :*

1. *$\Omega$ est la partie entre les éléments de $\Delta_G(\Omega)$.*
2. *$G_{\Delta_G(\Omega)}[\Omega]$ est une clique.*

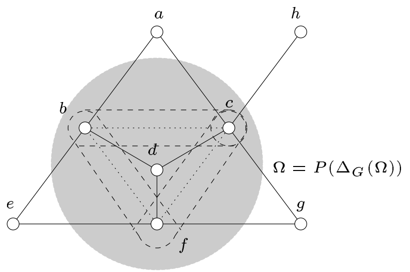


FIG. 3.6: Clique maximale potentielle : deuxième cas

*Preuve.* La preuve est très ressemblante à celle du théorème précédent.

"$\Rightarrow$" Considérons une triangulation minimale $H$ de $G$ dans laquelle $\Omega$ est une clique maximale. Par la proposition 3.8, on a $\Delta_H(\Omega) = \Delta_G(\Omega)$. Selon le théorème 3.3, $\Omega = P(\Delta_G(\Omega))$ dans $H$. Comme les composantes connexes associées à chaque

$S \in \Delta_G(\Omega)$ sont les mêmes dans $H$ et dans $G$, on a clairement que la partie entre les éléments de $\Delta_G(\Omega)$ est la même dans les deux graphes.

Enfin d'après la proposition 3.10, $\Omega$ est une clique dans $G_{\Delta_G(\Omega)}$.

"$\Leftarrow$" Remarquons que puisque $P(\Delta_G(\Omega))$ existe, les séparateurs minimaux de $\Delta_G(\Omega)$ sont deux à deux parallèles. Comme dans le théorème précédent, nous allons montrer que $\Omega$ est une clique maximale dans toute triangulation minimale $H$ de $G$ avec $\Delta_G(\Omega) \subseteq \Delta_H$. Clairement $\Omega$ est une clique de $H$ par la deuxième condition. Soit $x$ un sommet adjacent dans $H$ à tout sommet de $\Omega$. Encore une fois, $\Delta_H(\Omega) = \Delta_G(\Omega)$ par la proposition 3.6 et la partie entre les éléments de $\Delta_G(\Omega)$ est la même dans $G$ et dans $H$. Montrons que $\Omega$ est une clique maximale de $H$. Remarquons que, pour tout $S \in \Delta_G(\Omega)$, comme $\Delta_G(\Omega)$ n'a pas d'élément maximum pour l'inclusion, $\Omega \neq S$. Il existe donc un unique bloc $(S, C(S))$ contenant $\Omega$. Comme $x$ est adjacent dans $H$ à tout sommet de $\Omega$, en particulier à ceux de $\Omega \backslash S$, on doit avoir $x \in (S, C(S))$. Nous concluons, par définition de la partie entre une famille de séparateurs, que $x \in P(\Delta_G(\Omega))$, et donc $\Omega$ est une clique maximale de $H$. $\diamond$

Les théorèmes 3.11 et 3.12 fournissent un moyen de reconnaître une clique maximale potentielle. La chose la plus importante est que si l'on a un graphe $G$ et l'ensemble de ses séparateurs $\Delta_G$, ces deux théorèmes nous donnent un *algorithme* pour savoir si un ensemble de sommets forme une clique maximale potentielle. Ceci nous avait permis dans [BT98a] de calculer toutes les cliques maximales potentielles pour certaines classes de graphes. A ce jour nous n'avons pas d'algorithme énumérant toutes les cliques maximales potentielles d'un graphe quelconque en un temps polynomial en la taille du graphe et le nombre de ces cliques potentielles, comme on sait le faire pour les séparateurs minimaux.

Notons aussi que l'on peut reconnaître directement si une famille de séparateurs $\mathcal{S}$ est une famille maximale de séparateurs voisins. Si la famille possède un élément $S$ maximum par inclusion, il suffit de vérifier si pour un des blocs $(S, C)$ de $S$ dans $G$, l'ensemble de sommets $(S, C)$ est une clique maximale potentielle et si $\mathcal{S} = \Delta_G(S \cup C)$. Si $\mathcal{S}$ n'a pas de maximum, il faut que $P(\mathcal{S})$ existe, que ce soit une clique maximale potentielle de $G$ et enfin nous devons avoir $\mathcal{S} = \Delta_G(P(\mathcal{S}))$.

Si nous avons utilisé pour ces familles le terme de "séparateurs voisins", c'est parce que aucun autre séparateur minimal ne peut s'intercaler entre eux sans les croiser. Ils sont d'une certaine façon compacts, proches les uns des autres. Si l'on regarde le graphe de la figure 3.7 (les arêtes du graphe sont marquées en continu), les séparateurs minimaux $\{a, b\}$, $\{a, c\}$, $\{a, d\}$, $\{b, c\}$, $\{b, d\}$ et $\{c, d\}$ forment une famille maximale de séparateurs voisins et bordent la clique potentielle $\{a, b, c, d\}$. Ceci est quelque part en contradiction avec la notion habituelle de proximité que l'on a dans un graphe, on imagine mal en quoi les séparateurs $\{a, d\}$ et $\{b, c\}$ sont voisins. C'est peut-être ce qui explique en partie la difficulté que nous avons à trouver un algorithme d'énumération des cliques maximales potentielles et des familles de séparateurs voisins. Nous avons vraiment besoin de tous les séparateurs de la famille pour comprendre s'ils sont voisins (dans notre sens) ou pas.

Comme dans le cas des graphes triangulés, une famille maximale de séparateurs

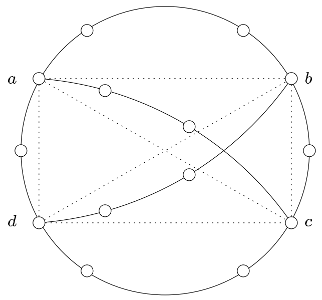


FIG. 3.7: Séparateurs voisins

voisins ayant un plus grand élément peut border plusieurs cliques maximales.

Par la suite, nous aurons besoin de calculer les cliques potentielles maximales pour des larges classes de graphes. Remarquons que les cliques potentielles “dégénérées”, de type $\Omega = (S, C)$, sont faciles à calculer pour les graphes quelconques. Il suffit pour cela de prendre un sommet quelconque et de vérifier si son voisinage fermé $N(x) \cup \{x\}$ est de la forme $(S, C)$ avec $x \in C$. Beaucoup plus surprenant, Bordat et Berry [BB98, BB96], qui travaillent avec une notion très proche de notre clique potentielle dégénérée, prouvent que l'on peut trouver une telle clique en temps linéaire. Plus exactement, ils montrent que tout graphe non complet a au moins deux cliques potentielles de ce type et que LexBFS a comme invariant le fait que les derniers sommets parcourus sont exactement les sommets de la composante connexe $C$ d'une clique potentielle $(S, C)$.

## 3.3 Une autre caractérisation des cliques maximales potentielles

Un des nos objectifs principaux est de calculer toutes les cliques maximales potentielles pour de larges classes de graphes. Comme nous l'avons dit, nous ne sommes pas encore en mesure de le faire pour tous les graphes ayant un nombre polynômial de séparateurs. L'énumération s'avérant difficile, nous avons besoin de bien comprendre ces objets et de les identifier de la façon la plus simple possible.

Dans cette section nous donnons une nouvelle caractérisation, très simple, des cliques maximales potentielles. Nous n'aurons plus besoin d'utiliser explicitement les séparateurs minimaux et surtout nous n'aurons plus les deux situations différentes, comme pour les théorèmes 3.11 et 3.12. Ces résultats nous permettront entre autres de calculer les cliques potentielles des graphes faiblement triangulés.

Comme précédemment, nous ferons des observations sur les cliques maximales des graphes triangulés et nous montrerons que certaines propriétés s'étendent aux cliques maximales potentielles.

Dans un graphe triangulé $H$, un séparateur $S$ contenu sans une clique maximale $\Omega$ est strictement inclus dans cette clique (cf. théorème 2.20), et de plus, $\Omega$ est dans un bloc plein de $S$ dans $H$. Avec les théorèmes 3.11 et 3.12 on constate aisément qu'il en est de même dans les graphes quelconques, comme le montre la proposition suivante :

**Proposition 3.13** *Soit $\Omega$ une clique maximale potentielle d'un graphe $G$ et soit $S \in \Delta_G(\Omega)$. Alors $S$ est strictement contenu dans $\Omega$ et $\Omega \backslash S$ est dans une composante connexe pleine associée à $S$ dans $G$.*

*Preuve.* Soit $H$ une triangulation minimale de $G$ telle que $\Omega$ soit une clique maximale de $H$. D'après la proposition 3.6, $S$ est un séparateur minimal de $H$. Comme $S$ est intersection de deux cliques maximales de $H$, il ne peut pas être lui-même une clique maximale. Il s'ensuit que $S$ est strictement contenu dans $\Omega$.

$\Omega$ étant une clique de $H$, elle doit se trouver dans un bloc de $S$ d'après le lemme 2.27. Comme tout élément de $S$ est adjacent dans $H$ à tout sommet de $\Omega \backslash S$, ce bloc est plein. Mais les blocs pleins de $S$ sont les mêmes dans $G$ que dans $H$, donc $\Omega$ est contenu dans un bloc plein associé à $S$ dans $G$. $\diamond$

Nous avons dit que, dans une clique maximale potentielle $\Omega$, la famille maximale de séparateurs voisins *borde* $\Omega$. En effet, elle sépare $\Omega$ du reste du graphe, comme l'exprime le lemme qui suit :

**Lemme 3.14** *Soient $\Omega$ une clique maximale potentielle de $G$ et $a$ un sommet de $V \backslash \Omega$. Il existe alors un séparateur minimal $S \subset \Omega$ de $G$ qui sépare $a$ et $\Omega \backslash S$.*

*Preuve.* Si nous sommes dans le cas du théorème 3.11, $\Delta_G(\Omega)$ a un élément maximum $S$ et on a $\Omega = (S, C)$. Clairement, $S$ sépare tout sommet $a \in V \backslash \Omega$ de tout sommet de $C = \Omega \backslash S$. Supposons maintenant que nous sommes dans le cas du théorème 3.12, i.e. $\Delta_G(\Omega)$ n'a pas d'élément maximum et $\Omega = P(\Delta_G(\Omega))$. Par définition de $P(\Delta_G(\Omega))$, comme $a \notin P(\Delta_G(\Omega))$, on déduit qu'il existe un $S \in \Delta_G(\Omega)$ séparant $a$ et $P(\Delta_G(\Omega)) \backslash S$. $\diamond$

Nous voulons caractériser une clique maximale potentielle $\Omega$ simplement en fonction des composantes connexes de $G \backslash \Omega$. Commençons par introduire quelques notations.

Soit $K$ un ensemble de sommets d'un graphe $G$. Nous allons noter par $C_1(K), \ldots,$ $C_p(K)$ les composantes connexes de $G \backslash K$. Pour chaque $i, 1 \leq i \leq p$, nous posons $S_i(K)$ l'ensemble des sommets de $K$ ayant au moins un voisin dans $C_i(K)$. Nous parlerons simplement de $S_i$ et $C_i$ lorsqu'il n'y a pas de risque de confusion sur l'ensemble $K$. Rappelons que si $S_i(K) = K$ on dit que $C_i(K)$ est une *composante pleine* associée à $K$.

**Proposition 3.15** *Soit $\Omega$ une clique maximale potentielle de $G$ et soit $\Delta_G(\Omega)$ la famille maximale de séparateurs voisins bordant $\Omega$. Les éléments de $\Delta_G(\Omega)$ sont exactement les ensembles $S_i(\Omega)$.*

*Preuve.* Prouvons que pour tout $i, 1 \leq i \leq p$, l'ensemble de sommets $S_i$ est un $a,b$-séparateur minimal, pour un couple de sommets $a$ et $b$. Par définition, $S_i$ est un séparateur de $G$. Comme $C_i \neq \emptyset$, nous pouvons choisir un sommet $a \in C_i$. Le lemme 3.14 nous dit qu'il existe un séparateur minimal $S \in \Delta_G(\Omega)$ qui sépare $a$ et $\Omega \backslash S$ ;

rappelons que $\Omega\backslash S$ est non vide, par la proposition 3.13. On a $S_i \subseteq S$ car s'il existait un sommet $x \in S_i\backslash S$, $x$ ayant au moins un voisin dans $C_i$, $S$ ne séparerait pas $x$ et $a$. Par la proposition 3.13, $\Omega\backslash S$ est dans une composante pleine associée à $S$, et donc à $S_i$. Soit $b$ un sommet quelconque de la composante de $G\backslash S$ contenant $\Omega\backslash S$. Clairement $a$ et $b$ sont dans des composantes connexes pleines distinctes induites par $S_i$, donc $S_i$ est un $a, b$-séparateur minimal par la proposition 2.14.

Il nous faut encore prouver que pour tout séparateur minimal $S \subset \Omega$, il existe un $i, 1 \leq i \leq p$, tel que $S = S_i$. Nous savons que $\Omega\backslash S \neq \emptyset$ et que $\Omega\backslash S$ est dans une composante connexe pleine associée à $S$. Soit $C$ une autre composante pleine associée à $S$. Remarquons que $C$ est aussi une composante connexe de $G\backslash\Omega$, mettons $C_i$. Comme tout sommet de $S$ a un voisin dans $C_i$, il s'ensuit que $S \subseteq S_i$. Supposons qu'il existe un $x \in S_i\backslash S$. Comme $C_i$ est une composante connexe pleine par rapport à $S_i$, $x$ a un voisin dans $C_i$ donc il doit être dans la composante connexe de $G\backslash S$ contenant $C_i$. On aurait donc $x \in C$, contredisant le fait que $C = C_i$. Il reste que $S = S_i$. On conclut que les séparateurs de $G$ contenus dans $\Omega$ sont exactement les ensembles $S_i$. $\diamond$

**Remarque 3.16** *En fait, comme $C_i$ est une composante connexe de $G\backslash S_i$, chaque séparateur minimal $S_i$ sépare tout sommet de $C_i$ de tout sommet de $G\backslash(S_i, C_i)$.*

Nous pouvons maintenant généraliser la proposition 3.15 aux cliques maximales potentielles et obtenir, en plus, une condition suffisante pour qu'un ensemble de sommets forme une clique maximale potentielle.

**Théorème 3.17** *Soit $K \subseteq V$ un ensemble de sommets du graphe $G$. On appelle $\mathcal{S}$ l'ensemble de tous les $S_i(K)$. $K$ est une clique maximale potentielle si et seulement si*

- *$K$ n'induit pas de composante connexe pleine dans $G$.*
- *$G_{\mathcal{S}}[K]$ est une clique.*

*De plus, si $K$ est une clique potentielle maximale, alors $\mathcal{S}$ est la famille maximale de séparateurs voisins bordant $K$.*

*Preuve.* Montrons l'implication "seulement si", qui découle assez directement de la proposition 3.15. Supposons que $K$ soit une clique maximale potentielle de $G$. Par la proposition 3.15, la famille maximale de séparateurs voisins bordant $K$ est exactement $\mathcal{S}$. Par les théorèmes 3.11 et 3.12, $K$ est une clique dans le graphe $G_{\mathcal{S}}$. Il reste à montrer que $K$ n'induit pas de composante connexe pleine. Soit $C_i$ une composante connexe de $G\backslash K$. Par définition les sommets de $K$ ayant un voisin dans $C_i$ sont exactement les éléments de $S_i$. Comme $K$ est une clique maximale potentielle et $S_i$ est un séparateur minimal de $G$ inclus dans $K$, on a que $S_i$ est strictement contenu dans $K$ par la proposition 3.13. Comme aucun élément de $K\backslash S_i$ n'a de voisin dans $C_i$, on en déduit que $C_i$ n'est pas une composante connexe pleine associée à $K$.

La preuve de l'autre implication est relativement longue. Nous vérifierons que les ensembles $S_i$ sont bien des séparateurs minimaux, qu'ils sont deux à deux parallèles et que $\Delta_G(K) = \mathcal{S}$. Enfin, nous prouverons que $K$ et $\mathcal{S}$ satisfont les conditions des théorèmes 3.11 ou 3.12.

Montrons au début que chaque $S_i, 1 \leq i \leq p$ est un séparateur minimal. Notons que $S_i$ est strictement contenu dans $K$, car $K$ n'induit pas de composante pleine d'après la première condition de l'hypothèse. $S_i$ est clairement un séparateur, car $C_i$ est une composante pleine associée à $S_i$ et $K \backslash S_i$ n'est pas contenu dans $C_i$, donc $G \backslash S_i$ a plusieurs composantes connexes. Soit maintenant $x$ un sommet de $K \backslash S_i$. Notons $C_x$ la composante connexe de $G \backslash S_i$ contenant $x$ et montrons qu'elle est pleine par rapport à $S_i$. Il nous faut prouver que chaque sommet $y \in S_i$ a au moins un voisin dans $C_x$. Ceci est vrai si $x$ et $y$ sont adjacents dans $G$. Dans le cas contraire, par la deuxième condition du théorème, $x$ et $y$ appartiennent à un même ensemble $S_j$. Puisque $C_j$ est une composante pleine associée à $S_j$, il existe une chaîne de $x$ à $y$ contenue dans $C_j$, excepté pour $x$ et $y$ (voir figure 3.8). Nous en déduisons que $C_j \subseteq C_x$. Il s'ensuit que $y$ a un voisin dans $C_x$, car il a un voisin dans $C_j$. On en conclut que $S_i$ induit deux composantes connexes pleines $C_x$ et $C_i$, donc il est un séparateur minimal conformément à la proposition 2.14. Ceci prouve que $\mathcal{S} \subseteq \Delta_G(K)$.

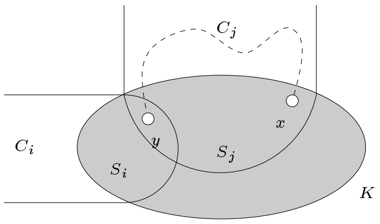


FIG. 3.8: $G_{\mathcal{S}}[K]$ est une clique

Maintenant, étant donnés deux séparateurs minimaux distincts $S_i$ et $S_j$, nous devons montrer qu'ils sont parallèles. Nous allons prouver que $K \backslash S_i$ est contenu dans une composante connexe de $G \backslash S_i$. Soit $x, y \in K \backslash S_i$. Si $x$ et $y$ sont adjacents, ils sont clairement dans la même composante connexe de $G \backslash S_i$. Autrement, comme $G_{\mathcal{S}}[K]$ est une clique, ils sont dans un même $S_k$, donc ils sont reliés via $C_k$. Donc $S_j$ intersecte seulement la composante connexe de $G \backslash S_i$ contenant $K \backslash S_i$, d'où $S_i \| S_j$. On en conclut que $\mathcal{S}$ est un ensemble de séparateurs minimaux deux à deux parallèles.

Nous devons montrer que $\Delta_G(K) \subseteq \mathcal{S}$, en d'autres termes que tout séparateur minimal de $G$ contenu dans $K$ est un élément de $\mathcal{S}$. Soit $H$ une triangulation minimale de $G$ telle que tous les éléments de $\mathcal{S}$ soit des séparateurs minimaux de $H$. Une telle triangulation existe, car les séparateurs minimaux $\mathcal{S}$ sont deux à deux parallèles d'après le point précédent. Nous savons que $K$ est une clique dans $H$. Soit maintenant $U \subseteq K$ un séparateur minimal quelconque de $G$. Remarquons que $U$ est strictement contenu dans $K$, sinon $K$ induirait deux composantes connexes pleines dans $G$, contredisant la première condition du théorème. Comme $U$ est une clique dans $H$, par la proposition 3.6 $U$ est aussi un séparateur minimal de $H$. Comme $K$ est une clique dans $H$, $K$ doit être contenu dans un bloc plein de $U$ dans $H$. Soit $(U, C)$ un autre bloc plein de $U$ dans $H$, et donc aussi dans $G$. Nous avons que $C$ est une composante connexe de $G \backslash U$ et $U$ sépare $C$ et $K \backslash U$. On en déduit que $C$

est aussi une composante connexe de $G\backslash K$, mettons $C_i$. Par définition de $S_i$, nous avons $U \subseteq S_i$. Supposons qu'il existe un sommet $x \in S_i\backslash U$. Comme $x$ a un voisin dans $C_i$ la composante connexe $C$ de $G\backslash U$ contiendrait $x$, contredisant $C = C_i$. Il reste que $U = S_i$ et $U \in \mathcal{S}$.

Pour finir nous allons vérifier que $K$ satisfait les conditions des théorèmes 3.11 ou 3.12. Nous savons que $\Delta_G(K) = \mathcal{S}$. Remarquons que pour tout $y \in V\backslash K$, $y$ est dans une certaine composante connexe $C_i$ de $G\backslash K$ et donc $S_i$ sépare $y$ de $K\backslash S_i$. Supposons maintenant que $\mathcal{S}$ ait un élément $S$ maximum pour l'inclusion. Soit $(S, C)$ le bloc de $S$ contenant $K$. Par la remarque précédente, pour tout $y \in V\backslash K$, $S$ sépare $y$ et $K\backslash S$, donc $y \notin (S, C)$. Il reste que $K = (S, C)$, donc par le théorème 3.11 $K$ est une clique maximale potentielle de $G$. Considérons maintenant le cas où $\mathcal{S}$ n'a pas d'élément maximum par inclusion. Clairement $K$ est contenu dans la partie entre les éléments de $\mathcal{S}$ dans $G$. Mais par la remarque précédente, $P(\mathcal{S})$ ne contient aucun sommet $y \in V\backslash K$, donc $K = P(\mathcal{S})$. Nous sommes dans les conditions du théorème 3.12. On conclut que $K$ est une clique maximale potentielle de $G$, bordée par $\mathcal{S}$. $\diamond$

Ce théorème nous donne un algorithme simple pour reconnaître une clique maximale potentielle :

**Corollaire 3.18** *Soit $K$ un ensemble de sommets du graphe $G$. Nous pouvons vérifier en temps $\mathcal{O}(n^3)$ si $K$ est une clique maximale potentielle de $G$.*

*Preuve.* L'algorithme 3.1 utilise le théorème 3.17 pour vérifier si l'ensemble de sommets $K$ est une clique maximale potentielle.

Le calcul des composantes connexes $C_i(K)$ de $G\backslash K$ et des ensembles $S_i(K)$ peut être fait en temps linéaire, tout en vérifiant si $K$ induit une composante connexe pleine. On complète ensuite chacun des $S_i$, ce qui coûte un temps de l'ordre de $|S_i|^2$, soit $\mathcal{O}(n^3)$ en tout. Nous obtenons ainsi le graphe $G_{\{S_1,\dots,S_p\}}$. Enfin, pour tout couple de sommets $x, y \in K$ on vérifie en temps constant si $x$ et $y$ sont reliés dans $G_{\{S_1,\dots,S_p\}}$.

Le coût total de l'algorithme est donc de $\mathcal{O}(n^3)$. $\diamond$

Comme par la suite nous tenterons d'énumérer les cliques maximales potentielles pour de larges classes de graphes, faisons quelques remarques simples sur le nombre de cliques potentielles par rapport au nombre de séparateurs minimaux d'un graphe.

D'après le théorème 3.17, une clique maximale potentielle contient au plus $n$ séparateurs minimaux. D'un autre côté, tout séparateur minimal $S$ est contenu en au moins une clique maximale potentielle. Pour s'en convaincre, il suffit de considérer une triangulation minimale $H$ de $G$ telle que $S$ soit un séparateur minimal de $H$ et de choisir une clique maximale de $H$ contenant $S$. Nous en déduisons que, si l'on note $p$ le nombre de cliques maximales potentielles de $G$, nous avons l'inégalité $p \geq |\Delta_G|/n$. En particulier, si le nombre de séparateurs minimaux d'un graphe est exponentiel en sa taille – comme dans le cas des grilles planaires carrées, par exemple – le nombre de cliques maximales potentielles de ce graphe est à son tour exponentiel. On en déduit :

---

**Algorithme 3.1 : Reconnaissance des cliques maximales potentielles**

---

`Entrée` : le graphe $G = (V, E)$ et l'ensemble de sommets $K \subseteq V$

`Sortie` : booléen indiquant si $K$ est une clique maximale potentielle

```
calculer les composantes connexes C_i de G\K
calculer les ensembles S_i
si il existe un i t.q. S_i = K alors
|   retourner "K n'est pas une clique potentielle"
pour i = 1 à p
|   compléter S_i
pour chaque couple de sommets x, y ∈ K
|   si x et y ne sont pas adjacents dans G_{S_1,...,S_p} alors
|   |   retourner "K n'est pas une clique potentielle"
retourner "K est une clique potentielle"
```

---

**Proposition 3.19** *Il n'éxiste pas d'algorithme énumérant les cliques maximales potentielles d'un graphe quelconque en temps polynomial par rapport à la taille du graphe.*

Nous ne savons pas à ce jour si tout graphe ayant un nombre polynomial de séparateurs minimaux a aussi un nombre polynomial de cliques maximales potentielles. C'est le cas pour toutes les classes de graphes que l'on trouve dans la littérature et qui ont peu de séparateurs minimaux, comme il sera montré dans le chapitre 5. De plus, nous ne savons pas non plus énumérer les cliques maximales potentielles en temps polynomial en leur nombre. Par contre, même les graphes planaires et les graphes sans triplet astéroïde ont un nombre de cliques maximales potentielles polynomial en le nombre de séparateurs minimaux. Nous n'avons pas à ce jour d'exemple de graphe avec plus de $\mathcal{O}(|\Delta_G|^2)$ cliques maximales potentielles.

## 3.4 Conclusion

Nous avons introduit les cliques maximales potentielles d'un graphe qui sont exactement les ensembles de sommets du graphe $G$ formant des cliques maximales dans les triangulations minimales $H$ de $G$. Nous verrons dans les chapitres suivants en quoi cet objet est un outil fondamental pour le calcul de la largeur arborescente et de la complétion minimale.

Le but de ce chapitre était de remplacer cette définition combinatoire par une caractérisation de nature algorithmique des cliques maximales potentielles. Nous avons vu que, étant donné un ensemble $K$ de sommets de $G$, nous pouvons vérifier facilement s'il s'agit d'une clique maximale potentielle.

Ceci peut être fait en regardant les séparateurs minimaux de $G$ contenus dans $K$, à l'aide des théorèmes 3.11 et 3.12. Une clique maximale potentielle est soit la partie entre les séparateurs $\Delta_G(K)$ qu'elle contient, soit un bloc du séparateur maximum par inclusion de $\Delta_G(K)$. Dans les deux cas, il suffit de compléter ces séparateurs et de vérifier si $K$ est une clique du graphe ainsi obtenu. Nous pouvons procéder plus simplement en regardant les composantes connexes de $G \backslash K$ et en utilisant le théorème 3.17. Il suffit de vérifier que $K$ n'induit pas de composante pleine et que, en complétant les ensembles $S_i(K)$, $K$ devient une clique.

Dans le chapitre 4 nous verrons que, si l'on dispose de toutes les cliques maximales potentielles d'un graphe, la largeur arborescente et la complétion minimale de ce graphe se calculent assez facilement. L'énumération des cliques maximales potentielles pour plusieurs classes de graphes fait l'objet du chapitre 5. Une clique maximale potentielle est en forte relation avec les séparateurs minimaux qu'elle contient. Pour certaines classes de graphes, pour lesquelles nous avons une bonne connaissance sur la structure des séparateurs, la première caractérisation des cliques potentielles donnée par les théorèmes 3.11 et 3.12 nous permettra d'énumérer ces objets. Pour d'autres classes de graphes, le théorème 3.17 nous aidera à identifier toutes les cliques maximales potentielles, même sans vraiment décrire comment sont structurés les séparateurs qu'elles contiennent.

# Chapitre 4

# Largeur arborescente et complétion minimale

Dans ce chapitre nous montrerons que, si nous savons calculer en temps polynomial toutes les cliques maximales potentielles d'une classe de graphes, nous pouvons calculer en temps polynomial la largeur arborescente et la complétion minimale de ces graphes.

Les algorithmes que nous allons donner ici prennent en entrée un graphe $G$, mais aussi les listes de ses séparateurs minimaux et des cliques maximales potentielles du graphe. Quant à l'énumération elle-même de toutes les cliques maximales potentielles pour certaines classes de graphes, elle fera l'objet du chapitre suivant.

Nous présenterons deux algorithmes différents. Le premier, légèrement plus rapide, ne peut être utilisé que pour le calcul de la largeur arborescente, plus exactement il décide si la largeur arborescente du graphe en entrée a au plus une valeur $k$. Le temps de calcul ne dépendant pas de $k$, on peut obtenir facilement un algorithme de calcul de la largeur arborescente en temps polynomial lorsque le nombre de cliques maximales potentielles est polynomial en la taille du graphe. Cet algorithme fonctionne en deux temps. D'abord, il vérifie si $\mathrm{tw}(G) \leq k$ et ensuite il donne une décomposition arborescente de $G$ de largeur au plus $k$. Il est à noter que toute triangulation minimale de ce type peut être obtenue par l'algorithme.

Le deuxième algorithme est plus "classique", généralisant des algorithmes calculant la largeur arborescente et la complétion minimale pour des classes de graphes traitées précédemment. C'est un algorithme de programmation dynamique sur les blocs.

Nous unifions ainsi les algorithmes précédents concernant le calcul de la largeur arborescente et de la complétion minimale pour certaines classes de graphes. Plus exactement, nous montrerons dans le chapitre 5 que tous les algorithmes déjà existants calculent en fait les cliques maximales potentielles des graphes respectifs. De plus, nous prouverons que le calcul de la largeur arborescente et de la complétion minimale est polynomial pour les graphes faiblement triangulés.

## 4.1 Calcul de la largeur arborescente

L'idée principale est de calculer d'abord toutes les cliques maximales potentielles $\Omega$ de $G$ qui interviennent dans des triangulations minimales de largeur au plus $k$. En d'autres termes, nous identifierons les cliques maximales potentielles $\Omega$ telles qu'il existe une triangulation minimale $H$ de largeur arborescente au plus $k$ et dans laquelle $\Omega$ est une clique maximale. Il faut donc commencer par identifier de manière simple ces cliques potentielles.

Soit $\mathcal{F} = \{\Omega_1, \Omega_2, \ldots, \Omega_p\}$ une famille de cliques maximales potentielles de $G$ et soit $\Gamma_{\mathcal{F}} = \{S | \exists \Omega \in \mathcal{F} \text{ t.q. } S \subset \Omega\}$ la famille de séparateurs minimaux de $G$ contenus dans au moins une de ces cliques potentielles. Nous dirons que $\mathcal{F}$ est une *famille complète de cliques maximales potentielles* si pour tout séparateur minimal $S \in \Gamma_{\mathcal{F}}$ et pour tout bloc plein $(S, C)$ de $S$, il existe une clique maximale potentielle $\Omega \in \mathcal{F}$ telle que $S \subset \Omega \subseteq (S, C)$. Une famille complète peut être vide, mais si ce n'est pas le cas, nous verrons que l'on peut en extraire une triangulation minimale du graphe dont toutes les cliques maximales sont des éléments de la famille.

En particulier, si $H$ est une triangulation minimale de $G$, la famille des cliques maximales de $H$ forme une famille complète de $G$. En effet, il nous suffit de prouver la propriété suivante :

**Lemme 4.1** *Soit $H$ un graphe triangulé et soit $(S, C)$ un bloc plein de $S$ dans $H$. Il existe une clique maximale $\Omega$ de $H$ telle que $S \subset \Omega \subseteq (S, C)$.*

*Preuve.* Les graphes triangulés étant héréditaires, le sous-graphe de $H$ induit par les sommets $S \cup C$ est triangulé. $S$ est une clique de $H[S \cup C]$ et $C$ est une composante connexe pleine associée à $S$ dans $H[S \cup C]$. Par le théorème 3.17, une clique maximale d'un graphe triangulé n'induit pas de composante connexe pleine. Il s'ensuit que $S$ n'est pas une clique maximale de $H[S \cup C]$. Soit $\Omega$ contenant $S$ une clique maximale de ce graphe. Comme $S$ sépare dans $H$ tout sommet de $\Omega \cap C$ de tout sommet de $V \backslash (S, C)$, $\Omega$ est une clique maximale de $H$. ◇

Remarquons que si $H$ est de largeur arborescente au plus $k$, toutes ses cliques auront au plus $k + 1$ sommets.

### 4.1.1 Description de l'algorithme

Nous commencerons par calculer la plus grande famille complète $\mathcal{F} = \{\Omega_1, \Omega_2, \ldots, \Omega_p\}$ de cliques maximales potentielles de $G$ telle que toute clique potentielle $\Omega \in \mathcal{F}$ ait au plus $k + 1$ sommets. Il s'agit d'un processus d'élimination, on veut repérer toutes les cliques potentielles maximales qui n'entrent pas dans la composition d'un arbre de cliques de largeur $k$.

**Algorithme d'élimination**

- `Entrée :` $G$, ses séparateurs minimaux et ses cliques maximales potentielles ; l'entier $k$.
- `Sortie :` la plus grande famille complète $\mathcal{F}$ de cliques maximales potentielles, dont tous les éléments ont au plus $k+1$ sommets.

1. Calculer toutes les cliques maximales potentielles de $G$ ayant au plus $k+1$ sommets. Soit $\mathcal{F}$ la famille de ces cliques potentielles et soit $\Gamma_{\mathcal{F}}$ l'ensemble des séparateurs minimaux contenus dans ces cliques.
2. Extraire de $\mathcal{F}$ la plus grande sous-famille complète de cliques potentielles :
   `tant que` il existe un $S \in \Gamma_{\mathcal{F}}$ et un bloc $(S, C)$ de $S$ tels qu'aucune clique potentielle $\Omega \in \mathcal{F}$ ne satisfait $S \subset \Omega \subseteq (S, C)$ `faire`
   - $\mathcal{F} = \mathcal{F} \backslash \{\Omega | S \subset \Omega\}$.
   - mettre à jour $\Gamma_{\mathcal{F}} = \{S | \exists \Omega \in \mathcal{F} \text{ tel que } S \subset \Omega\}$.
3. `retourner` $\mathcal{F}$.

Il est clair – par la définition même d'une famille complète – que la famille $\mathcal{F}$ de cliques maximales potentielles retournées par cette procédure est complète. Montrons qu'elle contient toutes les "bonnes" cliques potentielles maximales. Nous verrons plus loin qu'il s'agit bien de la plus grande famille complète de cliques maximales potentielles de taille au plus $k+1$.

**Lemme 4.2** *Soient $G$ un graphe et $H$ une triangulation minimale de $G$ telle que* $\mathrm{tw}(H) \leq k$*, i.e.* $\omega(H) \leq k+1$*. Si l'on applique à $G$ l'algorithme d'élimination, toute clique maximale de $H$ appartiendra à la famille $\mathcal{F}$ retournée par l'algorithme.*

*Preuve.* Soit $\mathcal{F}' = \{\Omega'_1, \ldots, \Omega'_q\}$ l'ensemble des cliques maximales de $H$. Chacune de ces cliques potentielles de $G$ ayant au plus $k+1$ sommets, elle appartiendra donc à $\mathcal{F}$ après le premier pas de l'algorithme.

Soit maintenant $S \in \Delta_H$ un séparateur minimal de $H$ et donc de $G$. Pour tout bloc $(S, C)$ de $S$, soit $\Omega$ une clique maximale de $H$ contenue dans $(S, C)$ et telle que $S \subset \Omega$. Une telle clique existe par le lemme 4.1, $S$ ne peut donc pas être choisi pour élimination. Il en résulte qu'aucune clique de $\mathcal{F}'$ ne peut être éliminée dans le deuxième pas de l'algorithme. À la fin de la procédure, nous avons bien $\mathcal{F}' \subseteq \mathcal{F}$. $\diamond$

Notons que ce résultat est vrai pour toutes les triangulations minimales $H$ telles que $\mathrm{tw}(H) \leq k$. Nous savons à ce stade que si $\mathrm{tw}(G) \leq k$, la famille de cliques retournées par l'algorithme d'élimination est non vide. La prochaine étape est de prouver que, en partant d'une famille complète non vide $\mathcal{F}$ de cliques maximales potentielles de $G$, on peut extraire une triangulation minimale $H$ telle que toute clique maximale de $H$ soit dans la famille $\mathcal{F}$. Nous en déduirons entre autres que toute famille complète dont toutes les cliques maximales potentielles ont au plus $k+1$ sommets est contenue dans la famille produite par l'algorithme d'élimination. Ceci prouve que l'algorithme d'élimination calcule l'unique plus grande famille complète de cliques maximales potentielles de cardinal au plus $k+1$.

En partant d'une famille complète de cliques maximales potentielles, nous allons construire directement un arbre $\mathcal{T} = (I, E_T)$, qui sera à la fin un arbre de cliques de $H$, et dont tous les nœuds seront étiquetés par des éléments de la famille. Au cours de l'algorithme, l'arbre aura deux types de nœuds :

- nœuds "traités", qui sont étiquetés par une clique maximale potentielle de $G$ appartenant à la famille $\mathcal{F}$.
- nœuds "non traités", auquel cas le nœud est une feuille de l'arbre et sera étiqueté par un bloc plein $(S, C)$, avec $S \in \Gamma_{\mathcal{F}}$.

Nous allons aussi étiqueter les arêtes de l'arbre par l'intersection des étiquettes des sommets adjacents, qui seront des séparateurs minimaux de $\Gamma_{\mathcal{F}}$.

---

**Algorithme d'extraction**

---

- `Entrée :` $G$ et une famille complète non vide $\mathcal{F}$ de cliques maximales potentielles.
- `Sortie :` un arbre de cliques $\mathcal{T}$ d'une triangulation minimale $H$ de $G$ telle que toutes les cliques maximales de $H$ soient des éléments de $\mathcal{F}$.

1. `Étape d'initialisation`
   - choisir une clique maximale potentielle $\Omega \in \mathcal{F}$.
   - créer la racine de $\mathcal{T}$ et l'étiqueter par $\Omega$.
   - marquer la racine comme "traitée".
   - `pour tout` bloc plein $(S, C)$ avec $S \subseteq \Omega$ et $\Omega \nsubseteq (S, C)$,
     - créer une feuille liée à la racine.
     - étiqueter la feuille par $(S, C)$ et l'arête adjacente par $S$.
     - marquer la feuille "non traitée".
   - mettre $\mathcal{F}' = \{\Omega\}$.
2. `Étape d'expansion`
   `tant qu'`il existe un noeud $N$ non traité `faire`
   - Soit $(S, C)$ l'étiquette de $N$. Choisir une clique maximale potentielle $\Omega \in \mathcal{F}$ telle que $S \subset \Omega \subseteq (S, C)$.
   - étiqueter le noeud $N$ par $\Omega$.

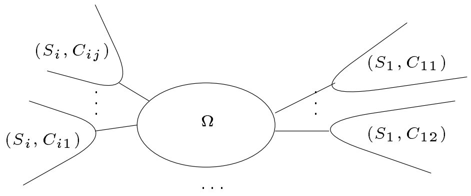


FIG. 4.1: Étape d'initialisation

  - `pour tout` bloc plein $(S', C')$ tel que $S' \subset \Omega$, $S' \not\subseteq S$ et $\Omega \not\subseteq (S', C')$ ,
    - créer une feuille $N'$ adjacente à $N$.
    - étiqueter $N'$ par $(S', C')$,
    - étiqueter l'arête $NN'$ par $S'$
    - marquer $N'$ comme non traité.
  - mettre $\mathcal{F}' = \mathcal{F}' \cup \{\Omega\}$.
  - marquer le noeud $N$ comme traité.

3. `retourner` $H = G_{\Gamma_{\mathcal{F}'}}$ et l'arbre étiqueté $\mathcal{T}$.

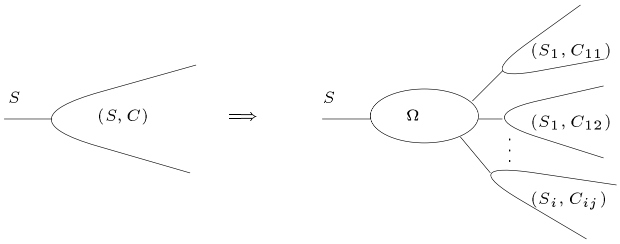


FIG. 4.2: Étape d'expansion

---

Remarquons d'abord que l'algorithme termine toujours. Lors de l'expansion d'un noeud étiqueté par $(S, C)$, les étiquettes $(S_i, C_i)$ des feuilles adjacentes seront strictement incluses dans $(S, C)$. En effet, si $\Omega$ est la nouvelle étiquette du noeud choisi, $\Omega \backslash S_i$ est non vide par la proposition 3.13 et par construction les sommets de $\Omega \backslash S_i$ ne sont pas dans le bloc $(S_i, C_i)$. On aura donc au plus $n$ étapes d'expansion, où $n$ est le nombre de sommets du graphe en entrée. Notons que lorsque l'on traite une feuille étiquetée $(S, C)$ où $(S, C)$ est une clique potentielle maximale de type dégénéré, la feuille gardera son étiquette et aucun noeud ne sera créé. En fait, toutes les feuilles de l'arbre final seront de ce type.

Comme invariants de l'algorithme, nous prouvons d'abord que tout sommet de $G$ se trouve dans une étiquette de $\mathcal{T}$. Nous verrons ensuite que si $(S, C)$ est une feuille non traitée de $\mathcal{T}$, alors $C$ n'intersecte aucune autre étiquette de $\mathcal{T}$. Enfin, si nous notons $\mathcal{T}'$ la partie traitée de l'arbre $\mathcal{T}$ et $V'$ l'ensemble de sommets de $G$ se trouvant dans des étiquettes de $\mathcal{T}'$, nous montrerons que $\Gamma_{\mathcal{F}'}$ est une famille de séparateurs deux à deux parallèles de $G$ et que $\mathcal{T}'$ est un arbre de cliques de $H = G_{\Gamma_{\mathcal{F}'}}[V']$. En fin de programme, $\mathcal{T}$ sera un arbre de cliques de la triangulation minimale $H = G_{\Gamma_{\mathcal{F}'}}$ de $G$.

**Lemme 4.3** *A tout moment de l'exécution, tous les sommets de $G$ sont dans au moins une étiquette de $\mathcal{T}$.*

*Preuve.* Après l'étape d'initialisation, tout sommet de $G$ se trouve soit dans la clique maximale potentielle $\Omega$ choisie comme racine soit dans l'une des composantes connexes $C_i$ de $G\backslash\Omega$. Dans ce dernier cas, nous savons par le théorème 3.17 que l'ensemble $S_i$ des sommets de $\Omega$ adjacents à un sommet de $C_i$ est un séparateur minimal et que $(S_i, C_i)$ est un bloc plein de $S_i$ dans $G$. Ceci nous garantit que $(S_i, C_i)$ sera l'étiquette d'une des feuilles adjacentes à la racine.

Considérons maintenant une étape d'expansion. Soit $(S, C)$ l'étiquette de la feuille non traitée choisie pour l'expansion et soit $\Omega$ la nouvelle étiquette du noeud. Encore une fois, tout sommet $x$ de $(S, C)$ est soit dans $\Omega$ soit dans une composante connexe de $(S, C)\backslash\Omega$. Mais les composantes connexes de $(S, C)\backslash\Omega$ sont aussi des composantes connexes de $G\backslash\Omega$, donc $x$ sera dans un bloc plein $(S_i, C_i)$ induit par $\Omega$ dans $G$. Prouvons que $S_i \not\subseteq S$. Par l'absurde, supposons que $S_i \subseteq S$. Dans ce cas $S$ sépare $C_i$ de $C\backslash C_i$ (remarquons que $C\backslash C_i$ n'est pas vide car il contient au moins $\Omega\backslash S$, non vide par la proposition 3.13). Ceci contredit le fait que $C$ est une composante connexe de $G\backslash S$. On en conclut que $S_i \not\subseteq S$, donc le bloc plein $(S_i, C_i)$ sera l'étiquette d'un noeud nouvellement créé. $\diamond$

**Lemme 4.4** *Soit $(S, C)$ l'étiquette d'une feuille non traitée de l'arbre $\mathcal{T}$. Alors $C$ n'intersecte aucune autre étiquette de $\mathcal{T}$.*

*Preuve.* Prouvons la propriété après l'étape d'initialisation. Soit $\Omega$ l'étiquette de la racine. Clairement $C \cap \Omega = \emptyset$ car $C$ est une composante connexe de $G\backslash\Omega$. Soit $(S', C')$ l'étiquette d'une autre feuille et supposons que $C$ intersecte $(S', C')$. Comme $C$ et $C'$ sont composantes connexes différentes de $G\backslash\Omega$, on a $C \cap C' = \emptyset$. Enfin, $S' \subset \Omega$, donc $C$ ne peut pas intersecter $S'$.

Considérons maintenant une étape d'expansion. Soit $N$ le noeud choisi pour l'expansion, d'étiquette $(S, C)$. Avant l'expansion, $C$ n'intersecte pas les sommets des autres étiquettes. Après l'expansion, soit $\Omega$ la nouvelle étiquette du noeud et soient $(S_i, C_i)$ les étiquettes des feuilles nouvellement créées. Comme pour le premier cas, aucun $C_i$ n'intersecte $\Omega$ ou une autre étiquette $(S_j, C_j)$ d'une feuille voisine de $N$. Mais $C_i \subset C$, donc $C_i$ ne peut pas intersecter une autre étiquette de l'arbre. $\diamond$

Notons $\mathcal{T}'$ le sous-arbre de $\mathcal{T}$ formé par les sommets traités, à tout moment de l'exécution. Posons $V' = \{x \in V | x \text{ est dans une étiquette de } \mathcal{T}'\}$. Notons simplement $\Gamma = \Gamma_{\mathcal{F}'}$ la famille des séparateurs minimaux de $G$ contenus dans une étiquette de $\mathcal{T}'$ et posons $H = G_\Gamma[V']$.

**Lemme 4.5** *A tout moment de l'exécution,* $\Gamma$ *est une famille de séparateurs deux à deux parallèles de* $G$ *et l'arbre étiqueté* $\mathcal{T}'$ *est un arbre de cliques de* $H = G_\Gamma[V']$.

*Preuve.* La propriété est évidente après l'étape d'initialisation.

Supposons que la propriété soit vraie à un certain moment de l'exécution et que nous traitions le noeud $N$ lors d'une étape d'expansion. Soient $(S, C)$ l'étiquette du noeud avant expansion et $\Omega$ la clique maximale potentielle choisie par le programme comme nouvelle étiquette de $N$. Nous notons par $\Gamma_e, V'_e, \mathcal{T}'_e$ et $H_e$ les paramètres après expansion.

D'abord, nous prouvons que les séparateurs de $\Gamma_e$ sont deux à deux parallèles. Clairement, $\Gamma_e = \Gamma \cup \Delta_G(\Omega)$. Les séparateurs de $\Delta_G(\Omega)$ sont parallèles, car contenus dans une même clique potentielle. Il suffit de montrer que si $T \in \Gamma$ et $U \in \Delta_G(\Omega)$ alors $T \| U$. Comme $T \subseteq V'$, $U \subseteq (S, C)$ et $C \cap V' = \emptyset$ (par le lemme 4.4), $S$ sépare $T$ et $U$, donc $T \| U$, par "pseudo-transitivité" de la relation de parallélisme (proposition 2.26).

Maintenant nous prouvons que $\mathcal{T}'_e$ est un arbre de cliques de $H_e = G_{\Gamma_e}[V'_e]$. Observons que $V'_e = V' \cup \Omega$. $\Omega$ est une clique de $H_e$, et elle est maximale car aucun sommet de $C \cap \Omega$ ne peut être adjacent à un sommet de $V' \backslash S$ par le lemme 4.4. De plus, la seule clique maximale de $H_e$ qui ne soit pas une clique maximale de $H$ est $\Omega$, car les arêtes de $H_e$ sont soit des arêtes de $H$ soit des arêtes de $H_e[\Omega]$. Les étiquettes de $\mathcal{T}'_e$ sont en conséquence exactement les cliques maximales de $H_e$. Il nous reste à prouver que $\mathcal{T}'_e$ satisfait la propriété d'intersection des cliques. Comme $\mathcal{T}'$ satisfait la propriété d'intersection des cliques par hypothèse de récurrence, il suffit de vérifier que pour toute étiquette $\Omega' \neq \Omega$ de $\mathcal{T}'_e$, l'ensemble $\Omega \cap \Omega'$ est contenu dans chaque clique sur la chaîne reliant $\Omega$ à $\Omega'$ dans $\mathcal{T}'_e$. $\Omega$ est une feuille de $\mathcal{T}'_e$, soit $\Omega_0$ l'unique voisin de $\Omega$ dans $\mathcal{T}'_e$. Tout ce qu'il nous faut montrer est que $\Omega \cap \Omega' \subseteq \Omega_0$. Or par le lemme 4.4, $\Omega \cap V' = S$, mais nous savons que $\Omega' \subseteq V'$ et que $S \subset \Omega_0$, donc $\Omega \cap \Omega' = S \cap \Omega' \subset \Omega_0$. $\diamond$

On déduit directement des lemmes 4.3 et 4.5 :

**Théorème 4.6** *Soient* $G$ *un graphe et* $\mathcal{F}$ *une famille complète de cliques maximales potentielles de* $G$. *Si* $\mathcal{F}$ *n'est pas vide, on peut construire une triangulation minimale* $H$ *de* $G$ *telle que toute clique maximale de* $H$ *soit un élément de la famille* $\mathcal{F}$.

*Preuve.* L'algorithme d'extraction fournit une triangulation $H$ de $G$ et un arbre de cliques $\mathcal{T}$ de $H$. Les cliques maximales de $H$ sont des étiquettes de $\mathcal{T}$, donc des éléments de $\mathcal{F}$.

Il nous reste à prouver que $H$ est une triangulation minimale de $G$. Tous les séparateurs de $H$ sont des intersections de cliques adjacentes de $\mathcal{T}$, donc $\Delta_H = \Gamma$. Le graphe $H$ est obtenu en complétant dans $G$ toutes les cliques potentielles $\Omega$ qui étiquettent $\mathcal{T}$, ce qui revient à compléter les séparateurs minimaux $\Delta_G(\Omega)$ de chacune de ces cliques, d'après la proposition 3.10. Par conséquent, $H = G_\Gamma$. Nous savons par le lemme 4.5 que $\Gamma$ forme une famille de séparateurs parallèles de $G$. Soit $S \in \Delta_G$ un séparateur minimal de $G$ parallèle à tous les éléments de $\Gamma$. D'après le lemme 2.29, $S$ est un séparateur minimal de $H$, donc la famille de séparateurs parallèles $\Gamma$ est maximale. Il s'ensuit que $H = G_\Gamma$ est une triangulation minimale de $G$. $\diamond$

Nous en concluons que, si l'on dispose de toutes les cliques maximales potentielles d'un graphe $G$, on peut décider si $\mathrm{tw}(G) \leq k$ en temps polynomial en le nombre de ces cliques potentielles et ne dépendant pas de $k$.

**Corollaire 4.7** *Soit $\mathcal{G}$ une classe de graphes pour laquelle on dispose d'un algorithme calculant en temps polynomial toutes les cliques maximales potentielles pour tout $G \in \mathcal{G}$. Alors pour tout graphe $G$ de $\mathcal{G}$ nous pouvons décider si $\mathrm{tw}(G) \leq k$, en un temps polynomial en la taille du graphe et ne dépendant pas de $k$.*

*Preuve.* Pour décider si la largeur arborescente d'un certain $G \in \mathcal{G}$ est au plus $k$, il suffit de :

1. appliquer l'algorithme d'élimination pour obtenir une famille complète $\mathcal{F}$ de cliques maximales potentielles ayant toutes au plus $k+1$ sommets.
2. `si` la famille $\mathcal{F}$ est vide `alors` $\mathrm{tw}(G) > k$.
3. `sinon` appliquer l'algorithme d'extraction sur $\mathcal{F}$ pour obtenir une triangulation minimale et une décomposition arborescente de largeur au plus $k$.

Par le lemme 4.2, si la largeur arborescente de $G$ est au plus $k$ l'algorithme d'élimination produit une famille $\mathcal{F}$ complète, non vide, de cliques maximales potentielles. Réciproquement, si la famille $\mathcal{F}$ issue de la phase d'élimination est non vide, d'après le théorème 4.6 l'algorithme d'extraction fournit une triangulation minimale et la décomposition arborescente associée, de largeur au plus $k$.

Clairement, toute ces étapes se font en un temps polynomial en $n$ et le nombre de cliques maximales potentielles de $G$, et la complexité de l'algorithme ne dépend pas de $k$. $\diamond$

Remarquons que, à partir de l'algorithme précédent qui décide si $\mathrm{tw}(G) \leq k$, on peut calculer aisément la largeur arborescente de $G$ en temps polynomial.

**Proposition 4.8** *Soit $\mathcal{G}$ une classe de graphes telle que pour tout $G \in \mathcal{G}$, on peut décider si $\mathrm{tw}(G) \leq k$ en un temps polynomial en $n$ et ne dépendant pas de de $k$. Alors la largeur arborescente de tout graphe de $\mathcal{G}$ est calculable en temps polynomial par rapport à la taille du graphe.*

*Preuve.* Pour tout graphe $G$, sa largeur arborescente varie entre 1 et $n-1$. En procédant par dichotomie sur l'intervalle $[1, n-1]$, nous calculons la largeur arborescente de $G$ en $\mathcal{O}(\log n)$ étapes de décision. Comme le coût d'une étape de décision est polynomial en $n$ et en la taille de $k$, le coût global de l'algorithme reste polynomial en $n$. $\diamond$

### 4.1.2 Implémentation et complexité

Pour implémenter cet algorithme, nous proposons la structure de données qui suit. Pour un accès plus rapide, chaque ensemble sera conservé par sa fonction caractéristique. Nous nous donnons en entrée la liste des séparateurs minimaux du graphe et la liste de ses cliques maximales potentielles, conservées elles aussi par leur fonction caractéristique.

1. chaque séparateur minimal $S$ connaît la liste de ses blocs pleins $(S, C)$. Pour chaque sommet $x$ de $G \backslash S$, nous garderons le bloc $(S, C_x)$ qui le contient.
2. chaque clique maximale potentielle $\Omega$ connaît la liste des séparateurs minimaux qu'elle contient, i.e. $\Delta_G(\Omega)$. Pour chaque $S \in \Delta_G(\Omega)$, on indique aussi le bloc $(S, C(S))$ contenant $\Omega$.
3. pour chaque bloc plein $(S, C)$, on garde la liste des cliques maximales potentielles $\Omega$ ayant la propriété $S \subset \Omega \subseteq (S, C)$.

Regardons tout d'abord le temps et l'espace nécessaires pour implémenter cette structure. Notons $r$ le nombre de séparateurs minimaux de $G$, $b$ le nombre de blocs et $p$ le nombre de cliques maximales potentielles. Le graphe en entrée a $n$ sommets et $m$ arêtes. Comme un séparateur minimal $S$ induit dans $G \backslash S$ au plus $n$ composantes, on a l'inégalité $b \leq rn$. Chaque ensemble que nous aurons à stocker (bloc, séparateur minimal, clique potentielle) a au plus $n$ sommets, donc chacun de ces objets nous coûtera un espace $\mathcal{O}(n)$.

La mémoire occupée par les $b$ blocs gardée par la fonction caractéristique est exactement $bn$. Calculer la liste des blocs pleins d'un séparateur minimal peut se faire en temps linéaire. Ceci nous coûte un temps de $\mathcal{O}(bn + rm)$ pour calculer tous les blocs. En même temps, nous indiquerons, pour chaque sommet $x \in G \backslash S$, le bloc $(S, C_x)$ de $S$ le contenant.

Pour que chaque clique $\Omega$ connaisse la liste des séparateurs minimaux $\Delta_G(\Omega)$ qu'elle contient, nous allons procéder de la façon suivante. Pour chaque clique potentielle $\Omega$ et pour chaque séparateur minimal $S$, nous vérifions si $S \subset \Omega$. Cette opération se fait en temps $\mathcal{O}(n)$, et nous récupérons par la même occasion un sommet $x \in \Omega \backslash S$, qui nous permettra d'identifier en temps constant le bloc de $(S, C(S))$ contenant $\Omega$. Ceci nous permet de mettre à jour les structures de données indiquées au points 2 et 3 en même temps. Le coût de l'opération pour une clique $\Omega$ et un séparateur $S$ est $\mathcal{O}(n)$, donc la complexité totale est $\mathcal{O}(prn)$.

Le coût global pour obtenir cette structure est donc $\mathcal{O}(prn + bn + rm)$ en temps et $\mathcal{O}((p + b)n)$ en espace.

Nous pouvons maintenant implémenter l'algorithme d'élimination en la même complexité. Il est plus simple de l'écrire de façon quelque peu différente que lors de sa présentation. Une clique potentielle qui doit faire partie de la solution, c'est-à-dire qui est une clique maximale d'une triangulation de largeur arborescente au plus $k$, est appelée *acceptable*. Les autres sont appelées *non acceptables*. Une clique acceptable $\Omega$ est dite *associée* au bloc $(S, C)$ si $S \subset \Omega \subseteq (S, C)$. Au début de l'algorithme toutes les cliques potentielles sont acceptables. Au cours de la procédure, certaines cliques $\Omega$ deviendront non acceptables et seront supprimées de la liste des cliques associées aux blocs $(S, C)$ avec $S \subset \Omega \subseteq (S, C)$. Lorsqu'un bloc $(S, C)$ n'a plus de clique associée, toutes les cliques potentielles contenant $S$ deviennent non acceptables.

La nouvelle version est illustrée par l'algorithme 4.1.

Cet algorithme fait exactement la même chose que la première version de l'algorithme d'élimination. A chaque étape, il ne fait que parcourir les pointeurs de la structure de données présentée plus haut. Il a une complexité linéaire par rapport à

**Algorithme 4.1 : Algorithme d'élimination - deuxième version**

`Entrée` : $G$, ses séparateurs minimaux et ses cliques maximales potentielles ; l'entier $k$

`Sortie` : la plus grande famille complète $\mathcal{F}$ de cliques maximales potentielles, dont tous les éléments ont au plus $k+1$ sommets

```
marquer toutes les cliques maximales potentielles comme acceptables
pour toute clique potentielle maximale Ω
    si |Ω| > k + 1 alors
        marquer Ω comme non acceptable non traitée
tant qu'il existe une clique potentielle Ω non acceptable non traitée
    pour tout bloc (S, C) avec S ⊂ Ω ⊆ (S, C)
        enlever Ω de la liste des cliques potentielles associées à (S, C)
        si la liste des cliques associées à (S, C) devient vide alors
            pour toute clique potentielle Ω' contenant S
                si Ω' est acceptable alors
                    la marquer comme non acceptable non traitée
    marquer Ω comme traitée
retourner la famille F des cliques acceptables
```

cette structure, il fonctionne donc en temps $\mathcal{O}(prn + bn)$.

Pour l'algorithme d'extraction, nous pouvons aussi obtenir du temps linéaire par rapport à notre structure de données. En effet, à chaque étape d'expansion on traite un nouveau bloc $(S, C)$, en choisissant une clique $\Omega$ associée à ce bloc en temps constant et en construisant les feuilles $(S', C')$. La construction des feuilles dépend linéairement du nombre de séparateurs minimaux de $\Delta_G(\Omega)$ et de leurs blocs. Chaque bloc $(S, C)$ est atteint deux fois, une fois lors de sa création du noeud étiqueté $(S, C)$ et une fois lorsque ce noeud est traité. Par conséquent, nous ne perdons pas en complexité par rapport à la taille de notre structure de données.

On en conclut que pour décider si $\mathrm{tw}(G) \leq k$ il nous faut un temps $\mathcal{O}(prn + bn + rm)$. Comme en général le nombre de cliques potentielles et de blocs est grand, disons au moins de l'ordre de $n$, nous pouvons majorer la complexité en temps par $\mathcal{O}(prn)$.

D'après la proposition 4.8, on peut transformer l'algorithme d'élimination, qui décide si $\mathrm{tw}(G) \leq k$, en un algorithme calculant la largeur arborescente de $G$. On utilise ensuite l'algorithme d'extraction pour obtenir une décomposition arborescente de largeur $\mathrm{tw}(G)$. Toutes ces opérations nous coûtent au plus $\mathcal{O}(prn \log n)$.

## 4.2 Algorithmes de programmation dynamique

Dans cette section nous donnons un algorithme de programmation dynamique qui calcule la largeur arborescente et la complétion minimale d'un graphe à partir de ses cliques maximales potentielles. Nous donnons d'abord une relation entre une triangulation minimale d'un graphe $G$ et certains blocs de $G$. Ceci nous conduira à des relations de récurrence pour calculer la largeur arborescente et la complétion minimale de ce que nous appellerons *réalisations* des blocs en fonction de blocs plus petits. Nous en déduirons un algorithme qui, par des techniques standard de programmation dynamique, calcule la largeur arborescente et la complétion minimale d'un graphe, à partir de ses cliques maximales potentielles.

### 4.2.1 Triangulations des blocs

Soit $(S, C)$ un bloc du graphe $G$. Le graphe $R(S, C) = G_S[S \cup C]$ est appelé la *réalisation* du bloc $(S, C)$. La proposition suivante, due à Kloks, Kratsch et Spinrad [KKS97] donne une relation entre les triangulations minimales d'un graphe et les triangulations minimales de certaines réalisations des blocs de $G$.

**Proposition 4.9** *Soit $S$ un séparateur minimal de $G$ et soient $C_1, \ldots, C_p$ les composantes connexes de $G \backslash S$. Soit $H_i$ une triangulation minimale de $R(S, C_i)$ pour tout $i, 1 \leq i \leq p$. Alors le graphe $H = (V, E(H))$ avec $E(H) = \bigcup_{i=1}^{p} E(H_i)$ est une triangulation minimale de $G$.*

*Réciproquement, soit $H$ une triangulation minimale de $G$ et soit $S$ un séparateur minimal de $H$. Alors $H[S \cup C]$ est une triangulation minimale de $R(S, C)$, pour tout bloc de $S$ dans $G$.*

*Preuve.* Prouvons la première partie du lemme. Tout d'abord, nous allons montrer que $H$ est un graphe triangulé. Supposons par l'absurde qu'il ait un cycle sans corde, de longueur au moins quatre. Ce cycle n'est pas contenu dans $S$ car $S$ est une clique dans $H$. Le cycle a donc un sommet $a$ dans une certaine composante $C_i$ de $G \backslash S$. Le cycle ne peut pas être contenu dans un $H_i$, car $H_i$ est triangulé. Il a donc un sommet $b$ dans une autre composante $C_j$ de $G \backslash S$. Comme $S$ sépare $a$ et $b$, les deux chaînes de $a$ à $b$ définies par le cycle coupent toutes les deux $S$, nous noterons $c$ et $d$ les sommets en question. Ils ne sont pas consécutifs sur le cycle. Mais comme $H[S]$ est une clique, $cd$ est une corde du cycle considéré.

Prouvons que $H$ est une triangulation minimale de $G$. Pour ce faire, considérons une arête $e \in E(H) \backslash E(G)$ et montrons que le graphe $H' = (V, E(H) \backslash \{e\})$ n'est pas triangulé. Il s'ensuivra d'après le lemme 2.8 que $H$ est une triangulation minimale de $G$. Vérifions que $H'$ a un cycle de longueur au moins quatre, sans corde. Notons que $S$ est un séparateur minimal de $H$, car toute composante connexe de $G \backslash S$ est aussi une composante connexe de $H \backslash S$, en particulier $S$ induit au moins deux composantes pleines dans $H$. Considérons une première situation où les deux extrémités $x$ et $y$ de $e$ sont dans $S$ et soient $C$ et $D$ deux composantes pleines associées à $S$ dans $H$. Prenons, dans le graphe $H$, des voisins $c_1$, respectivement $c_l$ de $x$, respectivement de $y$ dans $C$. Ces sommets existent car $C$ est une composante pleine associée à $S$. Nous pouvons alors construire une chaîne $[x, c_1, \ldots, c_l, y]$ reliant $x$ à $y$ dans $H$, avec tous les $c_i$ dans $C$. On peut prendre cette chaîne sans corde autre que $xy$. De la même manière, nous construisons une chaîne sans corde $[x, d_1, \ldots, d_q, y]$ avec $d_i \in D$. Ceci induit dans $H'$ le cycle $[x, c_1, \ldots, c_l, y, d_1, \ldots, d_q]$. Ce cycle a clairement au moins quatre sommets et n'a pas de corde, car $S$ sépare les $c_i$ et les $d_j$. Il s'ensuit que $H'$ n'est pas triangulé. Considérons maintenant le cas où les extrémités de $e$ ne sont pas toutes les deux dans $S$. Alors $e$ doit être l'arête d'un graphe $H_i$, et de plus $e \in E(H_i) \backslash E(R(S, C))$. Comme $H_i$ est une triangulation minimale de $R(S, C)$, $H_i$ privé de $e$ n'est pas un graphe triangulé, et c'est aussi vrai pour $H'$. Nous en déduisons que $H$ est une triangulation minimale de $G$.

Nous allons prouver la deuxième partie du lemme. Soit $H$ est une triangulation minimale de $G$ et soit $S$ un séparateur minimal de $H$. Soient $C_1, \ldots, C_p$ les composantes connexes de $G \backslash S$ et posons $H_i = H[S \cup C_i]$. Nous montrons que les $H_i$ sont des triangulations minimales de $R(S, C_i)$. Les $H_i$ sont des sous-graphes induits de $H$, ils sont donc triangulés. De plus, $S$ est une clique dans chacun des $H_i$, donc les $H_i$ sont des triangulations de $R(S, C_i)$. Prenons des triangulations minimales $H'_i \subseteq H_i$ de $R(S, C_i)$. D'après la première partie du lemme, le graphe $H' = (V, E(H'))$ avec $E(H') = \bigcup_{i=1}^{p} E(H'_i)$ est une triangulation minimale de $G$. Clairement $H'$ est un sous-graphe de $H$. Si un $H_i$ n'est pas triangulation minimale de $R(S, C_i)$, alors il existe une arête $e$ de $E(H_i) \backslash E(R(S, C_i))$ qui ne sera pas une arête de $E(H'_i)$. Comme au moins une extrémité de $e$ est dans $C_i$, $e$ n'appartient à aucun autre graphe $H_j$. Nous en déduisons que $E(H') \subset E(H)$, contredisant le fait que $H$ est une triangulation minimale de $G$. ◇

Ceci donne une équation pour calculer la largeur arborescente et la complétion minimale d'un graphe en fonction de la largeur arborescente et de la complétion minimale des réalisations de ses blocs :

**Corollaire 4.10** *Soit $G$ un graphe non complet. Alors*

$$\mathrm{tw}(G) = \min_{S \in \Delta_G} \max_{C \in \mathcal{C}(S)} \mathrm{tw}(R(S,C))$$

$$\mathrm{cm}(G) = \min_{S \in \Delta_G} (\mathrm{comp}(S) + \sum_{C \in \mathcal{C}(S)} \mathrm{cm}(R(S,C)))$$

*où* $\mathrm{comp}(S)$ *est le nombre d'arêtes de* $\overline{G}[S]$.

Nous donnons une version de la proposition 4.9 utilisant des cliques maximales potentielles à la place des séparateurs minimaux.

**Théorème 4.11** *Soient $H$ une triangulation minimale de $G$ et $\Omega$ une clique maximale de $H$. Pour chaque composante connexe $C_i, 1 \leq i \leq p$ de $G \backslash \Omega$, soit $S_i$ l'ensemble des sommets de $\Omega$ ayant un voisin dans $C_i$. Alors les sous-graphes $H_i = H[S_i \cup C_i]$ sont des triangulations minimales des réalisations $R(S_i, C_i)$.*

*Réciproquement, soit $\Omega$ une clique maximale potentielle de $G$. Pour chaque composante $C_i, 1 \leq i \leq p$ de $G \backslash \Omega$, soit $H_i$ une triangulation minimale de la réalisation $R(S_i, C_i)$. Alors $H = (V, E(H))$ avec $E(H) = \bigcup_{i=1}^{p} E(H_i) \cup \{xy | x, y \in \Omega\}$ est une triangulation minimale de $G$.*

*Preuve.* Pour prouver la première partie, remarquons que par le théorème 3.17, $(S_i, C_i)$ sont des blocs de $G$ et de $H$. Par la propriété 4.9, $H[S_i \cup C_i]$ sont des triangulations minimales des réalisations $R(S_i, C_i)$.

Prouvons la deuxième partie du lemme. Montrons que $H$ est un graphe triangulé. Supposons qu'il ait un cycle sans corde de longueur au moins quatre. Comme $\Omega$ est une clique, le cycle n'est pas inclus dans $\Omega$, donc il a un sommet $a$ dans une composante $C_i$. Puisque $H[S_i \cup C_i]$ est triangulé, le cycle n'est pas contenu dans $(S_i, C_i)$. Alors au moins deux sommets non consécutifs du cycle, mettons $b$ et $c$, sont dans $S_i$. Il s'ensuit que le cycle possède une corde, à savoir $bc$.

Montrons maintenant que $H$ est une triangulation minimale de $G$. Soit $e = xy$ une arête de $E(H) \backslash E(G)$. Nous voulons montrer que le graphe $H$ privé de $e$ n'est pas triangulé. Supposons d'abord que $x$ et $y$ ne sont pas tous les deux dans $\Omega$. Alors $x$ et $y$ sont dans un $H_i$. Comme $e$ n'est pas dans $\Omega$, elle n'est pas dans $S_i$. Donc $e$ est une arête de $E(H_i) \backslash E(R(S_i, C_i))$. Comme $H_i$ est une triangulation minimale de $R(S_i, C_i)$, par le lemme 2.8 le graphe $H_i$ privé de $e$ n'est pas triangulé, donc $H$ privé de $e$ n'est pas triangulé non plus. Considérons maintenant le cas $x, y \in \Omega$. Comme $H[\Omega] = G_{\Delta_G(\Omega)}[\Omega]$, les sommets $x$ et $y$ doivent être dans un même $S_i$. Prenons deux composantes connexes pleines induites par $S_i$ dans $H$ et une plus courte chaîne reliant $x$ et $y$ dans chacune d'entre elles. Les deux chaînes induisent un cycle de longueur au moins quatre, qui n'a pas de corde dans $H - e$. Par le lemme 2.8, on en déduit que $H$ est une triangulation minimale de $G$. ◇

Nous voulons donner une caractérisation des triangulations minimales d'une réalisation $R(S,C)$ utilisant les cliques maximales potentielles $\Omega$ avec $S \subset \Omega \subseteq (S,C)$ et des réalisations des certains blocs $(S_i,C_i)$ strictement contenus dans $(S,C)$. Ceci permettra d'exprimer la largeur arborescente et la complétion minimale d'une réalisation en fonction de la largeur arborescente, respectivement de la complétion minimale des réalisations de blocs plus petits. Les deux paramètres seront calculés par des techniques standard de programmation dynamique sur les blocs.

Les triangulations minimales des blocs non pleins sont exprimables facilement à partir d'un bloc plein inclus.

**Lemme 4.12** *Soient $(S,C)$ un bloc non plein de $S$ dans $G$ et $S'$ l'ensemble de sommets de $G$ ayant un voisin dans $C$. Alors $(S',C)$ est un bloc plein de $S'$ dans $G$ et un sur-graphe $H$ de $R(S,C)$ est une triangulation minimale de $R(S,C)$ si et seulement si $H[S' \cup C]$ est une triangulation minimale de $R(S',C)$.*

*Preuve.* Remarquons que $S'$ est un séparateur minimal de $R(S,C)$ et que $S = (S', S \backslash S')$ et $(S',C)$ sont les blocs pleins de $S'$ dans $R(S,C)$.

Si $H$ est une triangulation minimale de $R(S,C)$, le séparateur minimal $S'$ de $R(S,C)$ est une clique dans $H$, donc $S'$ est un séparateur minimal de $H$ d'après la proposition 3.6. On déduit directement de la proposition 4.9 que $H[S' \cup C]$ est une triangulation minimale du bloc $(S',C)$ de $R(S,C)$, or ce bloc est précisément $R(S',C)$.

Réciproquement, soit $H_1$ est une triangulation minimale de $R(S',C)$, c'est-à-dire du bloc $(S',C)$ de $R(S,C)$. Le bloc $S = (S', S \backslash S')$ est une clique dans $R(S,C)$, il est en conséquence sa propre triangulation minimale. D'après la proposition 4.9, tout sur-graphe $H$ de $R(S,C)$ tel que $H[S' \cup C] = H_1$ est une triangulation minimale de $R(S,C)$. $\diamond$

On en déduit directement :

**Corollaire 4.13** *Soit $(S,C)$ un bloc non-plein de $S$ dans $G$ et soit $S'$ l'ensemble des sommets de $S$ adjacents dans $G$ à au moins un sommet de $C$. Alors*

$$\mathrm{tw}(R(S,C)) = \max(|S| - 1, \mathrm{tw}(R(S',C)))$$

$$\mathrm{cm}(R(S,C)) = \mathrm{comp}(S) + \mathrm{cm}(R(S',C))$$

Il reste à exprimer la largeur arborescente et la complétion minimale d'un bloc plein en fonction des réalisations des blocs plus petits.

**Lemme 4.14** *Soit $R(S,C)$ la réalisation d'un bloc plein $(S,C)$ et soit $H(S,C)$ une triangulation de $R(S,C)$. Il existe alors une clique maximale $\Omega$ de $H(S,C)$ telle que $S \subset \Omega$ et $\Omega$ soit une clique maximale potentielle de $G$.*

*Preuve.* Le lemme est quasiment identique au lemme 4.1.

$S$ est une clique dans $H(S,C)$. Soit $\Omega$ une clique maximale de $H(S,C)$ contenant $S$. Alors $S \neq \Omega$, car $C$ est une composante pleine induite par $S$ dans $H(S,C)$ et par le théorème 3.17 une clique maximale d'un graphe triangulé n'induit pas de composante pleine. Il reste à montrer que $\Omega$ est une clique maximale potentielle de $G$. Par la proposition 4.9, il existe une triangulation $H$ de $G$ telle que $S$ soit un séparateur minimal de $H$ et $H[S \cup C] = H(S, C)$. Les composantes de $S$ sont les mêmes dans $H$ et dans $G$. Comme $S$ sépare $C$ de $V \backslash (C \cup S)$, $\Omega$ est une clique maximale de $H$ donc une clique maximale potentielle de $G$. $\diamond$

Observons que nous avons prouvé que toute clique maximale de $H(S,C)$ est une clique maximale potentielle de $G$ et, de plus, c'est une clique maximale dans toute triangulation minimale $H$ avec $H[S \cup C] = H(S,C)$.

Par la suite, $(S,C)$ est un bloc de $G$, et si $\Omega$ est une clique potentielle de $G$ avec $S \subset \Omega \subseteq (S,C)$, nous noterons $(S_i, C_i)$ les blocs engendrés par $\Omega$ dans $G$, contenus dans $(S,C)$. En d'autres termes, les $C_i$ sont les composantes connexes de $C \backslash \Omega$ et les $S_i$ sont les ensembles de sommets de $\Omega$ ayant des voisins dans $C_i$. Par le théorème 4.11 et le lemme 4.14 nous avons prouvé :

**Théorème 4.15** *Soit $(S,C)$ un bloc plein de $S$ dans $G$. Alors $H(S,C)$ est une triangulation minimale de $R(S,C)$ si et seulement si il existe une clique maximale potentielle $\Omega$ telle que $S \subset \Omega \subseteq (S,C)$ et $H(S,C) = (S \cup C, E(H))$ avec $E(H) = \bigcup_{i=1}^{p} E(H_i) \cup \{xy | x, y \in \Omega\}$ où les graphes $H_i$ sont des triangulations minimales de $R(S_i, C_i)$.*

Remarquons que les blocs $(S_i, C_i)$ sont strictement contenus dans $(S,C)$. En effet, pour tout $i$ le séparateur minimal $S_i$ est strictement contenu dans $\Omega$, et comme $\Omega \cap C_i = \emptyset$ nous en déduisons que les sommets de $(S,C)$ contenus dans $\Omega \backslash S_i$ ne se trouvent pas dans $(S_i, C_i)$.

**Corollaire 4.16** *Soit $(S,C)$ un bloc plein de $S$ dans $G$. Alors*

$$\mathrm{tw}(R(S,C)) = \min_{S \subset \Omega \subseteq (S,C)} \max_i (|\Omega| - 1, \mathrm{tw}(R(S_i, C_i)))$$

$$\mathrm{cm}(R(S,C)) = \min_{S \subset \Omega \subseteq (S,C)} \mathrm{comp}(S) + \sum_i \mathrm{cm}(R(S_i, C_i))$$

### 4.2.2 Implémentation et complexité

Si nous disposons des cliques potentielles d'un graphe $G$, les relations de récurrence des corollaires 4.13 et 4.16 nous permettent de calculer, pour tout bloc $(S,C)$ de $G$, la largeur arborescente et la complétion minimale de sa réalisation $R(S,C)$. À l'aide du corollaire 4.10 nous pourrons ainsi calculer la largeur arborescente et la complétion minimale de $G$.

L'implémentation de ces procédés se fait simplement par de la programmation dynamique sur les blocs. Ce deuxième algorithme est légèrement plus lent que le premier. Rappelons que nous avons noté $r, b$ et respectivement $p$ le nombre de séparateurs minimaux, de blocs et respectivement de cliques maximales potentielles de $G$. Nous utilisons presque la même structure de données que pour l'algorithme de calcul de la largeur arborescente donné dans la section précédente. Maintenant nous avons besoin de tous les blocs $(S,C)$ d'un séparateur minimal $S$, et non pas seulement de ses blocs pleins. De même, pour une clique potentielle maximale $\Omega$, nous gardons tous les blocs $(S_i, C_i)$ engendrés par $\Omega$. Cette structure nous coûte comme précédemment $\mathcal{O}(prn)$ en temps et $\mathcal{O}((p+b)n)$ en espace.

Nous commençons par trier les blocs en ordre croissant selon leur cardinal. Comme la taille des blocs est bornée par $n$, ceci se fait en temps $\mathcal{O}(b)$. Ensuite, on considère les blocs $(S, C)$ par ordre croissant de leur cardinal et on calcule la largeur arborescente et la complétion minimale de la réalisation $R(S, C)$. Clairement, lorsque l'on traite le bloc $(S, C)$, tous les blocs $(S', C')$ strictement inclus dans $(S, C)$ sont déjà traités, c'est-à-dire la largeur arborescente et la complétion minimale de $R(S', C')$ sont connus.

Si le bloc $(S, C)$ n'est pas plein on obtient $\mathrm{tw}(R(S, C))$ et $\mathrm{cm}(R(S, C))$ à l'aide du corollaire 4.13. Si le bloc $(S, C)$ est plein, pour chaque clique potentielle maximale $\Omega$ telle que $S \subset \Omega \subseteq (S, C)$, on prend les blocs $(S', C')$ engendrés par $\Omega$ et on applique les relations de récurrence du corollaire 4.16 afin d'obtenir $\mathrm{tw}(R(S, C))$ et $\mathrm{cm}(R(S, C))$ en fonction des blocs $(S', C')$. Pour un bloc $(S, C)$ et une clique potentielle $\Omega$ fixés, le temps dépend linéairement du nombre des blocs $(S', C')$ plus petits, qui peut être majoré par $n$. Globalement, l'algorithme de programmation dynamique coûte $\mathcal{O}(pbn)$ en temps, et $\mathcal{O}((p+b)n)$ en espace. Comme $b \leq rn$, on peut approximer la complexité en temps par $\mathcal{O}(prn^2)$.

On en conclut :

**Théorème 4.17** *Soit $\mathcal{G}$ une classe de graphes pour laquelle on dispose d'un algorithme calculant en temps polynomial toutes les cliques maximales potentielles pour tout $G \in \mathcal{G}$. Alors pour tout graphe de $\mathcal{G}$ la largeur arborescente et la complétion minimale sont calculables en temps polynomial.*

Notons que le facteur $n/\log n$ perdu par rapport au premier algorithme est dû au fait que pour un seul bloc plein $(S, C)$, nous devons considérer plusieurs cliques potentielles $\Omega$ avec $S \subset \Omega \subseteq (S, C)$, alors que l'étape d'expansion du premier algorithme nous permettait d'en choisir une seule.

## 4.3 Conclusion

Les algorithmes que nous venons de présenter prouvent que, si l'on dispose de toutes les cliques maximales potentielles d'un graphe $G$, alors le calcul de la largeur arborescente et de la complétion minimale de ce graphe se fait en un temps qui est polynomial en la taille du graphe et le nombre de cliques potentielles.

Comme on pouvait s'y attendre, ces algorithmes génériques ne sont pas vraiment efficaces. Nous verrons dans le chapitre suivant que, pour des classes de graphes particulières, il existe des algorithmes plus performants qui utilisent des propriétés structurelles particulières à chaque classe.

Nous avons prouvé que la connaissance des cliques maximales potentielles d'un graphe est suffisante pour calculer nos deux paramètres de triangulation. Nous montrerons dans le prochain chapitre que, pour toutes les classes que nous avons rencontré dans la littérature et pour lesquelles ces problèmes ont été résolus, les algorithmes respectifs calculent de manière implicite les cliques potentielles des graphes considérés.

En d'autres termes, les deux algorithmes de ce chapitre unifient les algorithmes déjà existants. Nous donnerons aussi un algorithme de calcul des cliques potentielles pour les graphes faiblement triangulés, pour lesquels le calcul de la largeur arborescente et de la complétion minimale était un problème ouvert.

Toutes ces considérations nous laissent penser que le calcul de la largeur arborescente et de la complétion minimale des graphes ayant un nombre polynomial de séparateurs passe par la connaissance des cliques maximales potentielles.

# Chapitre 5

# Application à des classes de graphes

Dans ce chapitre nous montrerons comment calculer toutes les cliques maximales potentielles pour certaines classes de graphes avec un nombre polynomial de séparateurs minimaux. Nous retrouverons toutes les classes de graphes pour lesquelles le calcul de la largeur arborescente et de la complétion minimale a déjà été résolu dans la littérature, du moins à notre connaissance. Enfin, nous donnerons un algorithme qui énumère les cliques maximales potentielles pour une nouvelle classe de graphes, à savoir les graphes faiblement triangulés.

On peut distinguer deux approches dans le calcul des cliques maximales potentielles. Pour certains graphes, nous avons une bonne connaissance de la structure des séparateurs minimaux. Nous pouvons ainsi décrire les cliques maximales potentielles à partir des séparateurs minimaux. C'est le cas notamment des graphes d'intersection comme les graphes de cordes ou d'intervalles circulaires ou encore des graphes sans triplet astéroïde.

Pour d'autres classes, nous pouvons caractériser directement les cliques maximales potentielles, sans forcément comprendre la structure des familles maximales de séparateurs voisins associés. C'est le cas des graphes HHD-free et bipartis cordaux, et plus généralement des graphes faiblement triangulés.

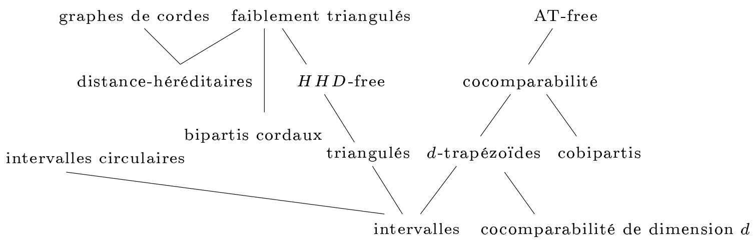


FIG. 5.1: Classes de graphes

La figure 5.1 illustre quelques classes de graphes que nous allons rencontrer dans ce chapitre, avec leur relations d'inclusion.

## 5.1 Graphes sans triplet astéroïde et leur généralisation

Nous commençons par une classe de graphes dont la structure des cliques maximales potentielles est très simple et découle presque immédiatement de leur définition. Les graphes *AT-free* ou *sans triplet astéroïde* n'ont pas en général un nombre polynomial de séparateurs minimaux. D'ailleurs, il a été prouvé par Arnborg, Corneil et Proskurowski [ACP87], respectivement par Yannakakis [Yan81] que le calcul de la largeur arborescente et de la complétion minimale sont des problèmes NP-complets pour les graphes co-bipartis, sous-classe des graphes sans triplet astéroïde.

Néanmoins, beaucoup de sous-classes des graphes AT-free, comme les graphes de permutation, les graphes trapézoïdes et $d$-trapézoïdes, les graphes de co-comparabilité de dimension bornée ont "peu" de séparateurs. Dans ce qui suit, nous décrirons les cliques maximales potentielles des graphes sans triplet astéroïde quelconques. Nous en déduirons que le calcul de la largeur arborescente et de la complétion minimale se fait en temps polynomial pour les graphes sans triplet astéroïde *ayant un nombre polynomial de séparateurs minimaux*. Des algorithmes calculant la largeur arborescente et la complétion minimale de ces graphes ont été donnés par Kloks, Kratsch et Spinrad [KKS97] et Parra [Par96].

La notion de triplet astéroïde est due à Lekkerkerker et Boland [LB62]. Ils l'utilisent pour caractériser les graphes d'intervalles comme étant exactement les graphes triangulés sans triplet astéroïde.

**Définition 5.1** *Trois sommets $(x, y, z)$ deux à deux non adjacents d'un graphe $G$ forment un* triplet astéroïde *si pour tout couple de sommets parmi $x, y$ et $z$ il existe une chaîne qui les relie sans intersecter le voisinage du troisième.*

*Un graphe* sans triplet astéroïde *est aussi appelé* AT-free.

De façon équivalente, nous pouvons dire que $x, y$ et $z$ forment un triplet astéroïde si $x$ et $y$ se trouvent dans la même composante connexe de $G\backslash N(z)$, $x$ et $z$ dans la même composante de $G\backslash N(y)$, et enfin $y$ et $z$ se trouvent dans la même composante de $G\backslash N(x)$.

Soit $G$ un graphe quelconque. Regardons ce qui se passe lorsque l'on considère une famille de trois séparateurs minimaux $\mathcal{S} = \{S, T, U\}$ de $G$, deux à deux parallèles et tels qu'aucun d'entre eux ne soit contenu dans un deuxième. Choisissons-les tels que pour chacun d'entre eux, il existe un bloc contenant les deux autres. En d'autres termes, la partie entre eux $P(\mathcal{S})$ existe, comme illustré sur la figure 5.2. Montrons que dans ce cas $G$ possède un triplet astéroïde.

Soit $(S, C(S))$ le bloc de $S$ contenant $T$ et $U$. Considérons un bloc plein $(S, C)$ de $S$, autre que $(S, C(S))$. En particulier, $(S, C)$ ne contient pas la partie entre $S$, $T$ et $U$. De la même façon, nous prenons les blocs pleins $(T, D)$ et $(U, E)$ ne contenant pas $P(\mathcal{S})$. Prenons des sommets $x, y$ et $z$ de $C, D$ et respectivement $U$. Nous allons montrer que $(x, y, z)$ est un triplet astéroïde. Soit $y'$ un sommet de $T\backslash S$. Comme $(T, D)$ est un bloc plein de $T$, il existe une chaîne de $y$ à $y'$ contenue dans $D \cup \{y'\}$.

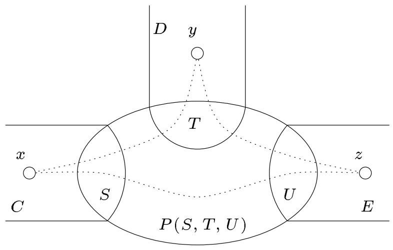


FIG. 5.2: Séparateurs minimaux et triplet astéroïde

Cette chaîne n'intersecte pas $S$, elle se trouve donc dans une composante connexe de $G\backslash S$. Nous en déduisons que $y$ est dans la composante connexe $C(S)$ du bloc $(S, C(S))$ contenant la partie entre les éléments de $\mathcal{S}$. Pour des raisons de symétrie, $z$ est dans le même bloc de $S$, donc $y$ et $z$ sont dans la même composante connexe de $G\backslash S$. Il s'ensuit qu'il existe une chaîne de $y$ à $z$ dans cette composante. Clairement, cette chaîne n'intersecte pas le bloc $(S, C)$ de $S$ contenant $x$, en particulier la chaîne n'intersecte pas le voisinage de $x$. Il existe donc une chaîne de $y$ à $z$ qui évite le voisinage de $x$. Il en est de même pour les autres couples de sommets de $x, y, z$ par rapport au troisième. On en conclut que $(x, y, z)$ est un triplet astéroïde de $G$.

Soient maintenant $G$ un graphe sans triplet astéroïde et $\Omega$ une clique maximale potentielle de $G$. Le nombre d'éléments maximaux par inclusion de la famille de séparateurs $\Delta_G(\Omega)$ est alors au plus deux. En effet, si $S$, $T$ et $U$ sont maximaux dans $\Delta_G(\Omega)$, alors $\Omega = P(\Delta_G(\Omega))$ est contenu dans $P(S, T, U)$. Ceci impliquerait que $G$ a un triplet astéroïde.

On en déduit :

**Proposition 5.2** *Soient $G$ un graphe sans triplet astéroïde et $\Omega$ une clique maximale potentielle de $G$. Alors la famille maximale de séparateurs voisins $\Delta_G(\Omega)$ possède au plus deux éléments maximaux par inclusion.*

*De plus, si $\Delta_G(\Omega)$ a deux éléments maximaux par inclusion $S_1$ et $S_2$, alors $\Omega = P(S_1, S_2)$.*

*Preuve.* Nous avons vu que $\Omega$ ne peut pas avoir trois éléments maximaux par inclusion ou plus. Il nous faut prouver la deuxième assertion.

Considérons une famille $\mathcal{S}$ de séparateurs minimaux d'un graphe $G$ quelconque telle que $P(\mathcal{S})$ existe et pour chaque $S \in \mathcal{S}$, le bloc $(S, C(S))$ contenant $P(\mathcal{S})$ est plein. Posons $\mathcal{S}' = \max(\mathcal{S})$ la famille des éléments maximaux de $\mathcal{S}$. Nous allons montrer que $P(\mathcal{S}) = P(\mathcal{S}')$. Remarquons que si $\Omega$ est une clique maximale potentielle n'ayant pas d'élément maximum par inclusion, la famille de séparateurs voisins $\mathcal{S} = \Delta_G(\Omega)$ satisfait les hypothèses plus haut.

Rappelons que $P(\mathcal{S}) = \bigcap_{S \in \mathcal{S}} (S, C(S))$, où $(S, C(S))$ est l'unique bloc de $S$ contenant tous les éléments de $\mathcal{S}$. Maintenant si $S, T \in \mathcal{S}$ et $T \subset S$, nous prouvons que le bloc $(S, C(S))$ est contenu dans le bloc $(T, C(T))$. Comme $C(S)$ est une composante connexe de $G\backslash S$ et $T \subset S$, $C(S)$ sera contenu dans une composante connexe de $G\backslash T$. Tout sommet $x \in S\backslash T$ a un voisin dans $C(S)$, car $C(S)$ est une composante pleine associée à $S$. Donc $x$ est dans la composante de $G\backslash T$ contenant $C(S)$. Il s'ensuit que $(S, C(S))$ est contenu dans un bloc de $T$. Comme $P(\mathcal{S}) \subseteq (S, C(S))$, le bloc de $T$ contenant $(S, C(S))$ est exactement le bloc $(T, C(T))$ contenant $P(\mathcal{S})$. Nous en déduisons que $(S, C(S)) \cap (T, C(T)) = (S, C(S))$. On en conclut facilement que $\bigcap_{S \in \mathcal{S}}(S, C(S)) = \bigcap_{S \in \mathcal{S}'}(S, C(S))$. $\diamond$

Une clique maximale potentielle $\Omega$ d'un graphe sans triplet astéroïde peut être donc de deux types : $\Omega = (S, C)$, où $S$ est le maximum par inclusion de $\Delta_G(\Omega)$, ou $\Omega = P(S, T)$, où $S$ et $T$ sont les deux éléments maximaux par inclusion de $\Delta_G(\Omega)$. Nous en déduisons qu'un graphe sans triplet astéroïde a au plus $|\Delta_G|(n + \frac{|\Delta_G|-1}{2})$ cliques maximales potentielles. Mais nous pouvons affiner nos remarques sur les cliques potentielles de deuxième type. Ces observations apparaissent dans la thèse de Parra [Par96].

Soit $\Omega = P(S, T)$ une clique maximale potentielle d'un graphe sans triplet astéroïde $G$, où $S$ et $T$ sont les éléments maximaux par inclusion de $\Delta_G(\Omega)$. Alors $S$ et $T$ sont des $a, b$-séparateurs minimaux pour un même couple de sommets $a$ et $b$. En effet, soit $C_a$ une composante pleine associée à $S$ différente de celle qui intersecte $\Omega$. On construit de la même façon une composante pleine $C_b$ associée à $T$ qui n'intersecte pas $\Omega$. Soient deux sommets $a \in C_a$ et $b \in C_b$. Soit $x$ un sommet de $S\backslash T$. Il existe une chaîne reliant $x$ à $a$ dans $C_a \cup \{x\}$. Comme ce chemin évite $T$, les sommets $x$ et $a$ sont dans la même composante connexe de $G\backslash T$, c'est-à-dire dans la composante pleine associée à $T$ intersectant $\Omega$. Il s'ensuit que $a$ et $b$ sont dans des composantes pleines distinctes associées à $T$ dans $G$, donc $T$ est un $a, b$-séparateur minimal par la proposition 2.14. De la même manière, $S$ est un $a, b$-séparateur minimal.

De plus, $S$ et $T$ sont des $a, b$-séparateurs consécutifs dans le treillis des $a, b$-séparateurs minimaux, au sens que nous avons évoqué dans la section 2.4 du chapitre 2. Plutôt que de développer cette idée, nous prouverons ici simplement que pour un séparateur minimal $S$ fixé, si l'on note $C$ l'une de ses composante connexes, alors il existe au plus $|C|$ séparateurs minimaux $T$ contenus dans le bloc $(S, C)$ tels que $P(S, T)$ soit une clique maximale potentielle de $G$. Soit $T \subseteq (S, C)$ un séparateur minimal tel que $\Omega = P(S, T)$ soit une clique potentielle. Soit $x$ un sommet de $S\backslash T$. Soit $C_b$ une composante connexe de $G\backslash T$ qui n'intersecte pas $\Omega$. Montrons que $C_b \subset C$. Prenons deux sommets $y \in T\backslash S$ et un $b \in C_b$. Comme $C_b$ est une composante pleine associée $T$, il existe une chaîne reliant $y$ à $b$ contenue dans $C_b \cup \{y\}$, donc cette chaîne n'intersecte pas $S$. Il s'ensuit que $y$ et $b$ sont dans la même composante connexe de $G\backslash S$, donc $C_b \subseteq C$. Considérons la réalisation $R(S, C)$ du bloc $(S, C)$. Rappelons que celle-ci est obtenue en complétant $S$ dans $G[S \cup C]$. On constate facilement que $T$ est un $x, b$-séparateur minimal dans $R(S, C)$, car $x$ et $b$ sont dans des composantes connexes pleines associées à $T$ dans $R(S, C)$. Comme $P(S, T)$ est une clique dans $G_{\{S,T\}}$, la partie $P(S, T)$ entre $S$ et $T$ sera une clique dans le graphe $R(S, C)_T$ obtenu à partir de de $R(S, C)$ en complétant $T$. Il s'ensuit que $x$ est adjacent à tout sommet de $T$ dans le graphe $R(S, C)$, donc $T \subseteq N_{R(S,C)}(x)$. Comme nous l'avons affirmé dans le premier chapitre (proposition 2.18 et remarque 2.19), pour tout couple de sommets $x, b$ d'un graphe, il existe un unique $x, b$-séparateur minimal contenu dans le voisinage de $x$. Par conséquent, $T$ est l'unique $x, b$-séparateur minimal de $R(S, C)$, contenu dans le voisinage de $x$.

Pour un séparateur $S$ fixé et pour un sommet $x \in S$ fixé, on trouve au plus $n$ sommets $b$ dans $G \backslash S$, donc au plus $T$ séparateurs minimaux de ce type, qui forment une clique potentielle avec $S$. En particulier, il existe au plus $n^2 |\Delta_G|$ couples de séparateurs minimaux $(S, T)$ formant une clique maximale potentielle $P(S, T)$, ce qui réduit à $\mathcal{O}(n^2 |\Delta_G|)$ le nombre de cliques maximales potentielles des graphes sans triplet astéroïde. Pour chaque séparateur minimal $S$ et pour chaque couple de sommets $x \in S$ et $b \in V(C) \backslash S$ on peut calculer en temps linéaire le $x, b$-séparateur minimal $T$ proche de $S$ et vérifier si $(S, T)$ forme une clique (cf. [Par96]). Il nous en coûtera donc un temps $\mathcal{O}(n^2 m |\Delta_G|)$ pour trouver les cliques maximales potentielles de type $P(S, T)$. D'un autre côté, les cliques maximales potentielles de type $(S, C)$ peuvent être énumérées en temps $\mathcal{O}(n^3 |\Delta_G|)$.

On en déduit :

**Proposition 5.3** *Si l'on dispose des séparateurs minimaux d'un graphe faiblement triangulé, ses cliques maximales potentielles peuvent être énumérées en un temps $\mathcal{O}(n^2 m |\Delta_G|)$. Le nombre de ces cliques est au plus $\mathcal{O}(n^2 |\Delta_G|)$.*

Il est à noter que toute triangulation minimale d'un graphe AT-free est un graphe d'intervalles, résultat dû à Möhring [Möh96]. La réciproque, due à Parra et Scheffler [PS97], est vraie aussi. Les graphes dont toutes les triangulations minimales sont des graphes d'intervalles sont précisément les graphes AT-free. Ceci fait que la largeur arborescente de ces graphes est égal à un autre paramètre, la largeur arborescente linéaire (pathwidth, en anglais). La largeur arborescente linéaire d'un graphe $G$ est définie de manière similaire à la largeur arborescente : elle est le minimum, parmi tous les sur-graphes d'intervalles $I$ de $G$, de $\omega(I) - 1$.

Les notions d'ensemble astéroïde peuvent s'étendre à plus de trois sommets.

**Définition 5.4** *Un ensemble de sommets $A$ d'un graphe $G$ quelconque est appelé* ensemble astéroïde *si, pour tout $a \in A$, les sommets $A \backslash \{a\}$ sont contenus dans une même composante connexe de $G \backslash N(a)$.*

*Le* nombre astéroïde *d'un graphe $G$, noté* $\mathrm{na}(G)$, *est le nombre maximum de sommets de $G$ qui forment un ensemble astéroïde.*

Cette notion est introduite par Walter [Wal78]. Les graphes sans triplet astéroïde peuvent donc être définis comme étant les graphes de nombre astéroïde égal à deux. Kloks, Kratsch et Müller prouvent dans [KKM97] que le calcul du nombre astéroïde est un problème NP-complet. Les mêmes auteurs et Broersma [BKKM98] donnent un algorithme pour calculer la largeur arborescente et la complétion minimale des graphes ayant peu de séparateurs et dont le nombre astéroïde borné par une constante.

On peut prouver de façon parfaitement identique aux graphes AT-free qu'une famille maximale de séparateurs voisins d'un graphe $G$ a au plus $\mathrm{na}(G)$ éléments maximaux. Si l'on considère la classes de graphes ayant le nombre astéroïde borné par une constante $c$, ceci fait que toute clique potentielle maximale d'un graphe triangulé est soit de type dégénéré $\Omega = (S, C)$, ou elle s'écrit sous la forme $\Omega = P(S_1, S_2, \ldots, S_p)$ ou $p \leq c$ et les $S_i$ sont les éléments maximaux par inclusion de $\Delta_G(\Omega)$. Ceci fait que si l'on sait qu'un graphe $G$ a un nombre astéroïde au plus égal à une constante $c$, ses cliques maximales potentielles peuvent être énumérées en temps $\mathcal{O}(n^3|\Delta_G|^c)$. Remarquons aussi que, d'après la définition du nombre astéroïde, on peut vérifier en temps $\mathcal{O}(n^{c+1}m)$ si $\mathrm{na}(G) \leq c$.

On en déduit :

**Proposition 5.5** *Etant donnés un graphe $G$, $\Delta_G$ l'ensemble de ses séparateurs minimaux et une constante $c$, on peut décider en temps $\mathcal{O}(mn^{c+1})$ si $G$ a un nombre astéroïde au plus $c$. Si oui, on peut énumérer ses cliques potentielles maximales en temps $\mathcal{O}(n^3|\Delta_G|^c)$.*

## 5.2 Graphes de cordes et graphes d'intervalles circulaires

Rappelons que les graphes de cordes (*circle graphs* en anglais) sont les graphes ayant comme modèle d'intersection une famille de cordes d'un cercle. Les graphes d'intervalles circulaires (*circular arc graphs* en anglais) ont comme modèle d'intersection une famille d'arcs de cercle.

Dans les deux cas, on peut supposer sans perdre en généralité que deux cordes distinctes, respectivement deux arcs distincts du modèle d'intersection d'un graphe de cordes, respectivement d'un graphe d'intervalles circulaires, n'ont pas d'extrémité commune. Notons aussi que, étant donné un graphe, on peut vérifier s'il appartient à une de ces classes et obtenir un modèle d'intersection en temps $\mathcal{O}(n^2)$ [ES93, Spi94]. Par la suite, chaque fois que nous utiliserons un graphe de cordes ou d'intervalles circulaires, nous supposerons qu'un modèle d'intersection de ce graphe est donné aussi.

Ces graphes sont d'une certaine façon très géométriques. Il est assez naturel de donner une interprétation géométrique aux séparateurs minimaux. Nous traitons d'abord en détail les graphes de cordes. Nous étendrons ensuite les résultats aux graphes d'intervalles circulaires.

Pour représenter géométriquement les séparateurs minimaux de ces graphes nous utiliserons la notion de *corde de contrôle*, introduite pour la première fois par Bodlaender et al. dans [BKK93]. Considérons un graphe de cordes $G$, avec un modèle d'intersection noté $\mathcal{D}(G)$. Les sommets de $G$ correspondent donc à des cordes de $\mathcal{D}(G)$ et deux sommets sont adjacents si et seulement si les cordes se croisent. Rappelons que deux cordes ne peuvent pas avoir une extrémité commune. Les extrémités des cordes forment un ensemble de points du cercle. Entre deux points consécutifs, nous plaçons un nouveau point, appelé *point de contrôle*. Chaque corde ayant comme extrémité deux points de contrôle est appelée *corde de contrôle* (*scanline*, en anglais). Soit $s$ une de ces cordes. Nous noterons $S(s)$ l'ensemble des cordes de $\mathcal{D}(G)$ qui intersectent $s$. Cette situation est illustrée par la figure 5.3, les cordes de $S(s)$ sont marquées en pointillés. Par abus de langage, $S(s)$ sera aussi l'ensemble des sommets de $G$ représentés par les cordes de $\mathcal{D}(G)$ croisant $s$.

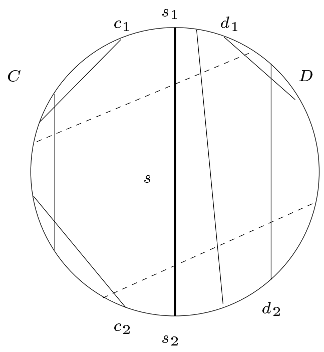


FIG. 5.3: Séparateur et corde de contrôle

Kloks caractérise les séparateurs minimaux de $G$ de la manière suivante :

**Théorème 5.6 ([Klo94, Klo96])** *Soit $G$ un graphe de cordes et $\mathcal{D}(G)$ un modèle d'intersection de $G$. Pour tout séparateur minimal $S$ de $G$, il existe une corde de contrôle $s$ telle que $S = S(s)$.*

*Preuve.* Considérons deux composantes connexes pleines $C$ et $D$ associées à $S$ dans $G$. Toute corde correspondant à un sommet de $C$ partitionne le cercle en deux régions, et les cordes de $D$ se trouvent dans la même région. Fixons un ordre de parcours du cercle, par exemple dans le sens des aiguilles d'une montre. Les cordes correspondant à $C$, respectivement $D$ sont séparées dans le modèle d'intersection, au sens où l'on peut trouver parmi les extrémités de ces cordes quatre points $c_1, d_1, d_2, c_2$, dans cet ordre, où $c_1$ et $c_2$ sont des extrémités de cordes de $C$, $d_1$ et $d_2$ sont des extrémités de cordes de $D$ et enfin toutes les extrémités des cordes de $C$, respectivement $D$ sont entre $c_2$ et $c_1$, respectivement $d_1$ et $d_2$ comme sur la figure 5.3.

Soit $s_1$ le premier point de contrôle entre $c_1$ et $d_1$ et $s_2$ le dernier point de contrôle entre $d_2$ et $c_2$. Montrons que la corde de contrôle $s = (s_1, s_2)$ satisfait $S = S(s)$. Pour ce faire il suffit de montrer que $S(s)$ est exactement l'ensemble des sommets $N(C)$ de $V(G)\backslash C$ ayant un voisin dans $C$. En effet, observons que toutes les cordes du modèle d'intersection qui croisent $s$ croisent aussi une corde de $C$. Il s'ensuit que $S(s) \subseteq N(C)$. Réciproquement, toute corde qui croise une corde de $C$ et qui n'est pas contenue dans $C$ intersecte la corde de contrôle $s$, d'où $N(S) \subseteq S(s)$. Nous en concluons que $S(s)$ est le voisinage $N(C)$ de $C$ dans $V(G)\backslash C$. Par définition d'une composante connexe pleine, cet ensemble est précisément $S$. $\diamond$

Le nombre de points de contrôle est $2n$ et le nombre de séparateurs minimaux de $G$ est majoré par le nombre de cordes de contrôle. En conséquence, les graphes de cordes ont un nombre polynomial de séparateurs.

**Corollaire 5.7** *Un graphe de cordes a au plus $n(2n-1)$ séparateurs minimaux.*

Nous avons prouvé que pour chaque séparateur $S$ minimal de $G$, il existe une corde de contrôle $s$ telle que les sommets de $S$ correspondent aux cordes du modèle qui croisent $s$. Remarquons que cette corde de contrôle a été définie par rapport à un couple de composantes connexes pleines $(C, D)$ associées à $S$ dans $G$ comme étant la corde de contrôle "la plus proche" des cordes de $C$, par rapport à $D$. Les cordes de $C$, respectivement $D$ se trouvent dans des régions différentes définies par $s$ dans le modèle d'intersection. Réciproquement, si deux cordes du modèle d'intersection sont séparées par $s$, alors $S(s)$ sépare dans $G$ les sommets respectifs. Il est facile de voir qu'à un séparateur minimal on peut associer en général plusieurs cordes de contrôle. Dans la suite, toutes ces cordes de contrôle seront déterminées par un couple de composantes connexes pleines induites par le séparateur représenté.

Pour calculer la largeur arborescente et la complétion minimale des graphes de cordes et des graphes d'intervalles circulaires, les auteurs des algorithmes respectifs donnent une caractérisation globale des triangulations minimales à l'aide des cordes de contrôle.

Ici nous donnerons directement une caractérisation des cliques maximales potentielles de ces graphes.

Nous dirons que deux cordes de contrôle $s$ et $t$ se croisent, noté $s \sharp t$, si elles ont une intersection non vide, autre qu'un point d'extrémité. En particulier, deux de ces cordes ayant une extrémité commune ne se croisent pas.

**Proposition 5.8** *Soit $G$ un graphe de cordes connexe. Soient $S$ et $T$ deux séparateurs minimaux de $G$. Considérons deux cordes de contrôle $s$ et $t$ correspondant à ces séparateurs. Alors les séparateurs $S$ et $T$ se croisent si et seulement si les cordes $s$ et $t$ se croisent.*

*Preuve.* Avant de faire la preuve, insistons sur le fait que les cordes de contrôle $s$ et $t$ sont choisies arbitrairement parmi les cordes de contrôle correspondant aux séparateurs $S$ et $T$. Toutefois, elles sont définies par rapport à un couple de composantes connexes pleines induites par $S$, respectivement $T$. La proposition reste vraie si $s$ et $t$ sont deux représentations d'un même séparateur.

Soit donc $s$ la corde de contrôle telle que $S = S(s)$, définie par rapport à un couple $(C, D)$ de composantes pleines associées à $S$ dans $G$ et telle que $s$ soit "proche" de $C$ au sens évoqué précédemment. Les cordes de $C$, respectivement $D$ se trouvent d'un côté et de l'autre de $s$. Si le séparateur $T$ croise $S$, il intersecte toutes les composantes connexes pleines induites par $S$ dans $G$ (proposition 2.25), en particulier $T$ intersecte $C$ et $D$. Comme $S(t) = T$, $t$ doit intersecter des cordes de $C$ et de $D$, donc $t$ croise $s$.

La réciproque est quelque peu plus délicate. Supposons que les cordes de contrôle $s$ et $t$ se croisent, mais que les séparateurs $S$ et $T$ soient parallèles. Notons $s_1, s_2$ les extrémités de $s$, de sorte que les extrémités des cordes de $C$ se trouvent sur l'arc de $s_2$ à $s_1$, toujours dans le sens des aiguilles d'une montre. Soient $t_1$ et $t_2$ les extrémités de $t$, où $t_1$ est sur l'arc de $s_2$ à $s_1$ et $t_2$ est sur l'arc de $s_1$ à $s_2$, comme sur la figure 5.4. Remarquons que puisque $t \sharp s$, $t$ intersecte au moins une corde de $C$. Notons $C'$, respectivement $D'$ les cordes du modèle d'intersection se trouvant du même côté de $s$ que $C$, respectivement $D$. Par conséquent, $C'$, $S$ et $D'$ forment une partition de $V(G)$ et $S$ sépare tout sommet de $C'$ de tout sommet de $D'$.

Comme $T = S(t)$, $T$ intersecte $C \subseteq C'$ et $T$ ne croise pas $S$, on en déduit que la

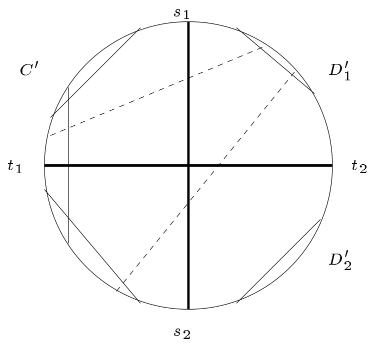


FIG. 5.4: Séparateurs et cordes de contrôle

corde de contrôle $t$ ne peut intersecter aucune corde de $D'$. En particulier, les cordes de $D$ sont dans une région définie par $s$ et $t$. Les extrémités des cordes de $D$ sont soit entre $s_1$ et $t_2$, soit entre $t_2$ et $s_2$. Sans perdre en généralité, supposons qu'elles soient entre $s_1$ et $t_2$. On partitionne $D'$ en deux, $D'_1$ contenant les cordes ayant leurs deux extrémités sur l'arc de cercle allant de $s_1$ à $t_2$ et $D'_2$ celles ayant leurs extrémités entre $t_2$ et $s_2$.

Nous avons partitionné le graphe $G\backslash S$ en trois parties $C'$, $D'_1$ et $D'_2$, et les cordes correspondantes se situent comme illustré la figure 5.4. Remarquons que, puisque $C$ et $D$ sont des composantes pleines induites par $S$ dans le graphe $G$, chaque corde correspondant à un sommet de $S$ intersecte une corde de $C$ et une corde de $D$. Par conséquent, chaque corde de $S$ aura une extrémité entre $s_2$ et $s_1$, et une autre entre $s_1$ et $t_2$. Il s'ensuit que les seules cordes ayant une extrémité dans la région contenant $D'_2$, sur l'arc de $t_2$ à $s_2$, sont précisément les cordes de $D'_2$. L'ensemble $D'_2$ n'est pas vide, car les points de contrôle $s_2$ et $t_2$ délimitant la région contenant $D'_2$ dont distincts. Comme aucune autre corde de $G$ n'a d'extrémité dans cette région, il s'ensuit que $D'_2$ est une partie de $G$ déconnectée du reste du graphe. Ceci contredit le fait que $G$ est connexe. On en conclut que les cordes de contrôle $s$ et $t$ ne se croisent pas. $\diamond$

Nous avons prouvé que les cordes de contrôle correspondant à deux séparateurs minimaux parallèles ne se croisent pas. Comme pour les séparateurs, nous dirons que deux cordes de contrôle qui ne se croisent pas sont parallèles. De même, si $\mathcal{S} = \{S_1, \ldots, S_p\}$ est une famille de séparateurs minimaux deux à deux parallèles, alors une famille de cordes de contrôle correspondant à ces séparateurs sera formée de cordes deux à deux parallèles.

Considérons maintenant une clique potentielle maximale $\Omega$ d'un graphe de cordes $G$. Soient $C_1, \ldots, C_p$ les composantes connexes de $G\backslash\Omega$ et soient $S_i$ les ensembles de sommets de $\Omega$ ayant un voisin dans $C_i$. Par le théorème 3.17, les $S_i$ sont des séparateurs minimaux de $G$. De plus, $C_i$ est une composante connexe pleine associée à $S_i$, et une autre composante pleine associée à $S_i$, notée $C(S_i)$, contient $\Omega\backslash S_i$. Pour chaque séparateur minimal $S_i$, considérons la corde de contrôle correspondant à $S_i$

qui est déterminée par le couple de composantes connexes pleines $(C_i, C(S_i))$. Les cordes sont donc choisies proches de $C_i$ et séparant $C_i$ de $C(S_i)$. Nous noterons $s(\mathcal{S}) = \{s_1, \ldots, s_p\}$ l'ensemble de ces cordes. Observons que l'on peut avoir deux séparateurs identiques $S_i$ et $S_j$, mais les cordes de contrôle $s_i$ et $s_j$ seront différentes.

Cette construction est illustrée par la figure 5.5, les cordes de contrôle dessinées en gras correspondent à la clique maximale potentielle $\Omega = \{a, b, c\}$. Dans ce cas, $\mathcal{S} = \{\{b, c\}, \{b\}, \{c\}\}$ et $\Omega$ est une clique de type dégénéré. Néanmoins, la structure des cordes de contrôle $s(\mathcal{S})$ est plutôt compliquée.

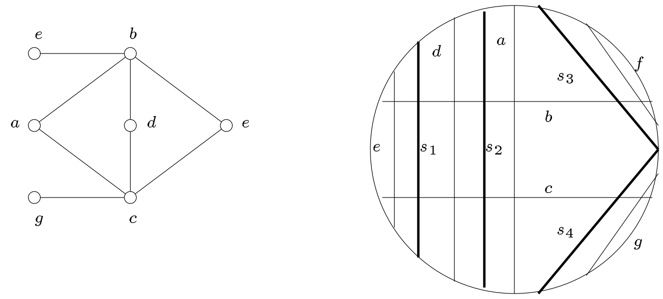


FIG. 5.5: Cliques potentielles des arcs d'intervalles circulaires

La caractérisation des cliques potentielles maximales est basée sur une observation très simple :

**Proposition 5.9** *Soit $\Omega$ une clique potentielle maximale d'un graphe de cordes $G$. Soit $s$ une corde de contrôle parallèle à toutes les cordes de $s(\mathcal{S})$. Alors les cordes de $\Omega$ intersectent toutes une des régions du cercle définies par $s$.*

*Preuve.* Supposons par absurde qu'il existent deux cordes $x$ et $y$ de $\Omega$ séparées par $s$. Alors les sommets $x$ et $y$ ne sont pas adjacents dans $G$. Comme $\Omega$ est une clique potentielle, il existe un séparateur minimal $S_i \in \mathcal{S}$ contenant $x$ et $y$, et comme $S_i = S(s_i)$ la corde de contrôle $s_i$ croise la corde de contrôle $s$, contredisant notre hypothèse. $\diamond$

Comme pour les séparateurs minimaux, un ensemble de cordes deux à deux parallèles et telles qu'aucune corde ne sépare deux autres induit une partie entre ces cordes. En particulier, nous considérerons des sous-ensembles de $s(\mathcal{S})$ avec cette propriété. Nous allons dire qu'un ensemble de cordes $\{s_{i_1}, \ldots, s_{i_q}\}$ contenu dans $s(\mathcal{S})$ forme une *région élémentaire* si la partie entre les cordes $\{s_{i_1}, \ldots, s_{i_q}\}$ existe et ne contient aucune corde de $s(\mathcal{S})$, en d'autres termes aucune corde de $s(\mathcal{S})$ n'intersecte l'intérieur de la région. Une région élémentaire pourra être aussi un côté d'une corde $s_i \in s(\mathcal{S})$ tel qu'aucune autre corde de $s(\mathcal{S})$ ne se trouve à l'intérieur.

A l'aide de la proposition 5.9, nous prouverons que les cordes représentant les sommets de $\Omega$ sont exactement les cordes intersectant une certaine région élémentaire définie par les cordes de contrôle de $s(\mathcal{S})$.

**Proposition 5.10** *Soit $\Omega$ une clique potentielle maximale d'un graphe de cordes $G$. Il existe alors une région élémentaire $\mathcal{R}$ formée par des cordes de contrôle de $s(\mathcal{S})$*

*telle que les cordes de $\Omega$ sont exactement les cordes du modèle intersectant la région $\mathcal{R}$.*

*Preuve.* Montrons d'abord que toutes les cordes de $\Omega$ intersectent une certaine région élémentaire. D'après la proposition 5.9, toutes les cordes de $\Omega$ intersectent un côté de $s_1$, que nous noterons $\mathcal{R}_1$. Si aucune corde de contrôle de $s(\mathcal{S})$ n'est contenue dans $\mathcal{R}_1$, alors $\mathcal{R}_1$ est la région élémentaire cherchée. Sinon, soit $s_i$ une corde contenue dans $\mathcal{R}_1$. Alors $s_i$ divise $\mathcal{R}_1$ en deux régions $\mathcal{R}_1'$ et $\mathcal{R}_1''$ et, de manière identique à la proposition 5.9, on déduit que toutes les cordes de $\Omega$ intersectent l'une de ces régions, par exemple $\mathcal{R}_1'$. Nous posons $\mathcal{R}_2 = \mathcal{R}_1'$ et nous recommençons à la diviser jusqu'à obtenir la région élémentaire $\mathcal{R}$ intersectée par toutes les cordes représentant les sommets de $\Omega$.

Il nous reste à prouver qu'aucune corde de $G\backslash\Omega$ n'intersecte $\mathcal{R}$. Soit $x$ un sommet dont la corde intersecte la région élémentaire $\mathcal{R}$. Observons que si la corde $x$ intersecte une des cordes de contrôle $s_i$ bordant la région $\mathcal{R}$, alors $x \in S(s_i)$, donc $x \in S_i \subset \Omega$. Il reste la possibilité que la corde $x$ soit complètement contenue dans $\mathcal{R}$. Supposons donc que $x \in G\backslash\Omega$, mais la corde $x$ est complètement contenue dans $\mathcal{R}$. Le sommet $x$ est dans une composante connexe $C_i$ de $G\backslash\Omega$, et $S_i$ sépare $x$ de tout sommet $y \in \Omega\backslash S_i$. La corde de $y$ intersecte $\mathcal{R}$ comme sur la figure 5.6. Comme la corde de contrôle $s_i$ sépare les cordes de $x$ et $y$, par construction de $s_i$, il s'ensuit que la corde de contrôle $s_i$ intersecte la région $\mathcal{R}$, contredisant le fait que cette région est élémentaire.

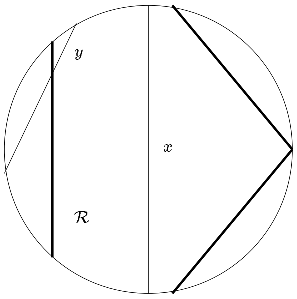


FIG. 5.6: Cordes croisant une région élémentaire

Nous en concluons que les cordes du modèle $\mathcal{D}(G)$ intersectant la région $\mathcal{R}$ sont exactement les cordes de $\Omega$. $\diamond$

Jusqu'ici nous n'avons utilisé que le modèle d'intersection et les séparateurs minimaux du graphe. Malheureusement, la proposition 5.10 ne nous donne pas de moyen algorithmique pour calculer les régions élémentaires et en conséquence les cliques potentielles maximales du graphe. Nous utiliserons un artifice pour contourner ce problème, mais le fait d'avoir recours à lui indique que nous ne savons pas calculer les cliques potentielles à partir simplement des séparateurs minimaux.

Reprenons la région élémentaire $\mathcal{R}$ telle que les cordes intersectant $\mathcal{R}$ soient exactement les cordes de $\Omega$. Nous allons trouver trois points de contrôle $p_1, p_2$ et $p_3$

dans $\mathcal{R}$, formant un triangle $\mathcal{R}_T \subseteq \mathcal{R}$, de sorte que toutes les cordes de $\Omega$ intersectent ce triangle.

**Théorème 5.11** *Soit $\Omega$ une clique potentielle maximale d'un graphe de cordes $G$. Il existe un triangle $\mathcal{R}_T$ défini par trois points de contrôle et tel que les cordes du modèle $\mathcal{D}(G)$ intersectant $\mathcal{R}_T$ sont exactement les cordes de $\Omega$.*

*Preuve.* Prenons une région élémentaire $\mathcal{R}$ de sorte que les cordes de $\Omega$ soient exactement les cordes intersectant $\mathcal{R}$. Si $\mathcal{R}$ est un triangle, l'assertion est prouvée. Sinon, nous allons procéder de la même façon que lors de la construction de la région $\mathcal{R}$. Soient deux points de contrôle de $\mathcal{R}$ formant une corde $s$ qui divise $\mathcal{R}$ en deux régions $\mathcal{R}'$ et $\mathcal{R}''$. Comme dans la proposition 5.9, $s$ ne peut pas séparer deux cordes de $\Omega$, donc toutes les cordes de $\Omega$ intersectent l'une des régions $\mathcal{R}'$ ou $\mathcal{R}''$. Nous réitérons le procédé jusqu'à trouver le triangle $\mathcal{R}_T$ voulu. $\diamond$

On en conclut :

**Corollaire 5.12** *Un graphe de cercle $G$ a $\mathcal{O}(n^3)$ cliques maximales potentielles, qui sont calculables en temps polynomial.*

*Preuve.* En effet, $G$ a au plus autant de cliques potentielles que de triangles de points de contrôle. Les points de contrôle sont en nombre de $2n$, il y a donc au plus $\mathcal{O}(n^3)$ cliques maximales potentielles.

Pour les calculer, il suffit de considérer chaque triangle de points de contrôle et de calculer l'ensemble des cordes qui l'intersectent. On vérifie ensuite si cet ensemble est une clique potentielle maximale. $\diamond$

Nous retrouverons ce résultat pour les graphes d'intervalles circulaires. Néanmoins, la preuve est quelque peu différente.

Les notions de point de contrôle et de corde de contrôle s'adaptent aussi aux graphes d'intervalles circulaires. Encore une fois, nous considérons tous les points d'extrémité des arcs de cercle d'un modèle d'intersection $\mathcal{D}(G)$ de $G$. Le modèle est choisi de sorte que deux arcs distincts ne partagent pas la même extrémité. Entre chaque couple d'extrémités consécutives $a$ et $b$ on choisit un point $p$. On décide que $p$ est un point de contrôle si et seulement si le nombre d'arcs du modèle d'intersection passant par $p$ est plus petit que le nombre d'arcs passant par $a$, respectivement par $b$ ; nous n'avons pas besoin de cette restriction dans nôtre cas, mais elle est utile par contre pour la caractérisation globale des triangulations minimales d'un graphe d'intervalles circulaires donnée dans [Kra96, KKW98]. Une corde entre deux points de contrôle est appelée corde de contrôle. Soit $s$ une corde de contrôle, d'extrémités $s_1$ et $s_2$. On note $S(s)$ l'ensemble des arcs du modèle d'intersection contenant au moins un des points $s_1$ ou $s_2$. Par abus de langage, on notera de la même manière l'ensemble des sommets correspondant à ces arcs.

Comme pour les graphes de cordes, les séparateurs minimaux d'un graphe d'intervalles sont représentés par des cordes de contrôle :

**Théorème 5.13 ([Kra96, KKW98])** *Soit $G$ un graphe d'intervalles circulaires et $\mathcal{D}(G)$ un modèle d'intersection de $G$. Pour tout séparateur minimal $S$ de $G$, il existe une corde de contrôle $s$ telle que $S = S(s)$.*

La preuve de ce théorème est tout à fait similaire à celle du résultat équivalent pour les graphes de cordes, nous ne la refaisons pas ici.

Il s'ensuit :

**Corollaire 5.14** *Un graphe d'intervalles circulaires a au plus $n(2n-1)$ séparateurs minimaux.*

Considérons maintenant un ensemble de sommets $K$ d'un graphe d'intervalles circulaires $G$. Pour chaque composante connexe $C$ de $G \backslash K$, on définit l'arc de cercle qui est l'union des arcs de $C$. Les composantes connexes de $G \backslash K$ induisent des arcs disjoints. Sur le cercle, les composantes $C_1, \ldots, C_p$ forment les arcs $(a_1, b_1) \ldots, (a_p, b_p)$, où $a_i$ et $b_i$ sont les extrémités de l'arc correspondant à $C_i$. On peut considérer que ces points se trouvent sur le cercle dans l'ordre $a_1, b_1 \ldots, a_p, b_p$, comme illustré par la figure 5.7. Posons $a_{p+1} = a_1$. Pour tout $i, 1 \leq i \leq p$, prenons entre $b_i$ et $a_{i+1}$ les points de contrôle $b'_i$ et $a'_{i+1}$ les plus proches de $b_i$, respectivement $a_{i+1}$. Ces points existent, on peut éventuellement avoir $b'_i = a'_{i+1}$. Les cordes de contrôle $(a'_i, b'_i)$ forment une région $\mathcal{R}$. Il est facile de voir que les sommets de $K$ correspondent exactement aux arcs intersectant cette région.

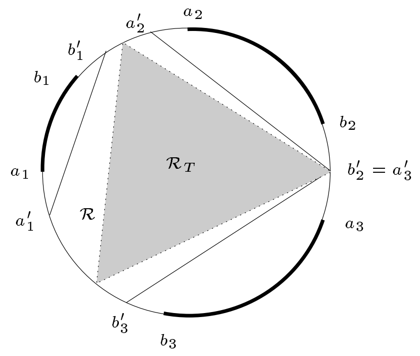


FIG. 5.7: Cliques potentielles des arcs d'intervalles circulaires

Maintenant, si $K = \Omega$ est une clique maximale potentielle de $G$, les cordes de contrôle $s_i$ d'extrémités $(a'_i, b'_i)$ correspondent aux séparateurs minimaux $S_i$. Comme nous l'avons fait pour les graphes de cordes, on prouve que pour toute corde de contrôle $s$ ne croisant pas les cordes $s_i$, tous les arcs de $\Omega$ intersectent un côté de $s$. Il s'ensuit que région élémentaire $\mathcal{R}$ déterminée par les cordes $s_i$ est telle que les sommets de $\Omega$ correspondent aux arcs intersectant $\mathcal{R}$. A partir de la région élémentaire $\mathcal{R}$, on montre qu'il existe un triangle $\mathcal{R}_T$ déterminé par trois points de contrôle et tel que les sommets de $\Omega$ correspondent exactement aux arcs contenant au moins un point de $\mathcal{R}_T$.

On en conclut :

**Théorème 5.15** *Soit $\Omega$ une clique potentielle maximale d'un graphe d'intervalles circulaires $G$. Il existe un triangle $\mathcal{R}_T$ formé par trois points de contrôle et tel que les arcs du modèle $\mathcal{D}(G)$ intersectant $\mathcal{R}_T$ sont exactement les arcs de $\Omega$.*

Comme pour les graphes de cordes, ceci nous permet de calculer en temps polynomial toutes les cliques potentielles d'un graphe d'intervalles circulaires :

**Corollaire 5.16** *Un graphe de cercle $G$ a $\mathcal{O}(n^3)$ cliques maximales potentielles, qui sont calculables en temps polynomial.*

Bien sur, les algorithmes de [SSR94, Klo96, KKW98] qui calculent la largeur arborescente et la complétion minimale des graphes de cordes et d'intervalles circulaires sont beaucoup plus performants que notre algorithme générique. Tous ces problèmes peuvent être résolus en $\mathcal{O}(n^3)$. Le principe de base de ces algorithmes est le suivant. On note $Z$ l'ensemble des points de contrôle d'un graphe de cordes ou d'intervalles circulaires. Ces points forment un polygone $P$. On considère une *triangulation planaire* de $P$, c'est-à-dire on rajoute à $P$ des diagonales qui ne se croisent pas de manière à le partitionner en triangles. Ces diagonales et les arêtes du polygone forment un ensemble de cordes de contrôle deux à deux parallèles, noté $T$. Pour une triangulation planaire $T$ de $P$, on note $H(T)$ le graphe obtenu à partir de $G$ en complétant les ensembles $S(s)$ pour chaque corde $s \in T$. En d'autres termes, $H(T) = G_{\{S(s) \mid s \in T\}}$. On peut prouver le théorème qui suit :

**Théorème 5.17 ([KKW98])** *Soit $G$ un graphe de cordes ou d'intervalles circulaires. Pour toute triangulation minimale $H$ de $G$, il existe une triangulation planaire $T$ du polygone des points de contrôle de $G$ telle que $H = H(T)$. Réciproquement, pour toute triangulation planaire $T$, $H(T)$ est un graphe triangulé.*

Les algorithmes déjà connus de calcul de la largeur arborescente et de la complétion minimale pour les graphes de cordes et d'intervalles circulaires n'utilisent pas exactement les triangulations minimales de ces graphes, mais une famille plus grande de triangulations, celles de type $H(T)$, comme dans le théorème 5.17. Ces algorithmes peuvent être implémentés en un temps qui dépend linéairement du nombre de triangles de points de contrôle, donc en temps $\mathcal{O}(n^3)$. Bien que nettement plus rapides que notre algorithme générique, ces algorithmes utilisent quand même toutes les cliques maximales potentielles du graphe en entrée.

## 5.3 Graphes faiblement triangulés

Il existe deux autres classes de graphes, non contenues dans celles déjà citées, pour lesquelles le calcul de la largeur arborescente et de la complétion minimale se fait en temps polynomial. Il s'agit des graphes bipartis cordaux et des graphes $HHD$-free. Nous ne les présentons pas ici, car nous donnons un algorithme polynomial pour calculer les cliques potentielles maximales des graphes faiblement triangulés, qui contiennent les deux classes de graphes mentionnés.

Il est à noter que l'on savait que les graphes faiblement triangulés ont un nombre polynomial de séparateurs [KBMK93]. Malgré cela, le problème du calcul de la largeur arborescente et la complétion minimale de ces graphes n'avait pas été résolu. Les résultats présentés dans cette section sont nouveaux. La technique utilisée pour le calcul des cliques potentielles des graphes faiblement triangulés est assez particulière, fortement inspirée de l'algorithme de reconnaissance de ces graphes. Néanmoins, elle est peut-être un point de départ pour le calcul des cliques potentielles pour tout graphe ayant un nombre polynomial de séparateurs.

Les graphes faiblement triangulés ont été introduits par Hayward dans [Hay85]. Il prouve notamment que cette classe contient les graphes triangulés et que les graphes faiblement triangulés sont parfaits.

**Définition 5.18** *Un graphe $G$ est dit* faiblement triangulé *si, à la fois dans $G$ et dans son complémentaire $\overline{G}$, tout cycle ayant au moins cinq sommets possède une corde.*

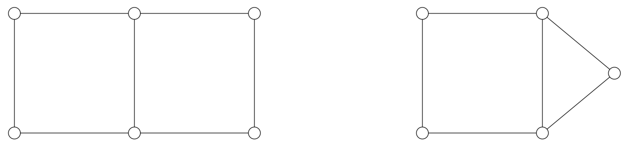

FIG. 5.8: Graphes faiblement triangulés

Les graphes de la figure 5.8 sont faiblement triangulés. Parmi les graphes faiblement triangulés on trouve aussi les graphes bipartis cordaux - c'est-à-dire bipartis sans cycle induit de longueur au moins 6 - et les graphes HHD-free - sans cycle induit de longueur au moins 6 et n'ayant pas les graphes de la figure 5.8 comme sous-graphes induits. Pour ces classes le calcul de la largeur arborescente et de la complétion minimale a été résolu par [KK95, Cha96], respectivement [BDK97a].

Une notion essentielle dans la reconnaissance des graphes faiblement triangulés est celle de *deux-paire* :

**Définition 5.19** *Deux sommets $x$ et $y$ forment une* deux-paire *si leurs voisins communs $N(x) \cap N(y)$ forment un $x,y$-séparateur.*

On voit que $x$ et $y$ forment une deux-paire si et seulement si toute chaîne sans corde reliant $x$ à $y$ a exactement deux arêtes. En particulier, $x$ et $y$ ne sont pas adjacents. Hayward, Hoàng et Maffray prouvent dans [HHM89] que si $G$ est un graphe faiblement triangulé, alors tout sous-graphe induit $G[W]$ qui n'est pas une clique contient une deux-paire. En particulier, si $G$ n'est pas une clique, il contient une deux-paire.

Spinrad et Sritharan [SS95] donnent un algorithme de reconnaissance des graphes triangulé basé sur le théorème suivant :

**Théorème 5.20** *Soit $G = (V, E)$ un graphe et soit $\{x, y\}$ une deux-paire de $G$. Notons $G' = (V, E \cup \{xy\})$ le graphe obtenu à partir de $G$ en lui rajoutant l'arête $xy$. Alors $G$ est faiblement triangulé si et seulement si $G'$ est faiblement triangulé.*

Remarquons qu'une clique est un graphe faiblement triangulé. L'algorithme 5.1 de reconnaissance des graphes faiblement triangulés se déduit très naturellement du théorème 5.20.

**Algorithme 5.1 : Algorithme de reconnaissance des graphes faiblement triangulés**

`Entrée` : Le graphe $G_0$

`Sortie` : booléen indiquant si $G_0$ est faiblement triangulé

soit $G = G_0$

`tant que` $G$ a une deux-paire $\{x, y\}$
    rajouter à $G$ l'arête $xy$

`si` $G$ est une clique `alors` $G_0$ est faiblement triangulé

`sinon` $G_0$ n'est pas faiblement triangulé

En effet, si $G$ est devenu une clique, le graphe initial était faiblement triangulé d'après le théorème 5.20. Si $G$ n'est pas une clique et n'a pas de deux-paire, $G$ n'est pas faiblement triangulé d'après les remarques précédentes. Il s'ensuit d'après le théorème 5.20 que le graphe en entrée n'est pas triangulé.

Nous montrerons que les graphes faiblement triangulés ont un nombre polynomial de séparateurs. Notons $\overline{m}$ le nombre d'arêtes du graphe $\overline{G}$. Soit $G_0 = (V, E)$ un graphe triangulé et soient $e_1 = x_1y_1, \ldots, e_{\overline{m}} = x_{\overline{m}}y_{\overline{m}}$ les arêtes ajoutées au graphe $G$ par l'algorithme de reconnaissance, dans cet ordre. Notons par $G_i$ le graphe $(V, E \cup \{e_1, e_2, \ldots, e_i\})$, pour $0 \leq i \leq \overline{m}$. En particulier $G_{\overline{m}}$ est une clique, $G_0$ est toujours le graphe en entrée. Nous allons décrire les séparateurs minimaux de $G_i$ en fonction des séparateurs minimaux de $G_{i+1}$. Ensuite, nous décrirons les cliques maximales potentielles de $G_i$ en fonction de celles de $G_{i+1}$.

**Théorème 5.21** *Soit $G$ un graphe faiblement triangulé non complet et soit $\{x, y\}$ une deux-paire de $G$. Posons $S_{xy} = N(x) \cap N(y)$ les voisins communs de $x$ et $y$. Nous notons $G'$ le graphe obtenu à partir de $G$ en rajoutant l'arête $xy$. Dans ces conditions, $\Delta_G \subseteq \Delta_{G'} \cup \{S_{xy}\}$.*

*Preuve.* $S_{xy}$ est un $x, y$-séparateur de $G$ par définition d'une deux-paire. Notons que tout $x, y$-séparateur contient les voisins communs de $x$ et $y$. $S_{xy}$ est donc un $x, y$-séparateur minimal de $G$. Comme tout $x, y$-séparateur contient $S_{xy}$, on en déduit que $S_{xy}$ est l'unique $x, y$-séparateur minimal de $G$.

Soit $S$ un autre séparateur minimal de $G$ et montrons que $S$ est aussi un séparateur minimal de $G'$.

Supposons d'abord que $S$ sépare $x$ et $y$ dans $G$ et soient $C_x$ et $C_y$ les composantes connexes de $G \backslash S$ contenant $x$, respectivement $y$. Remarquons que les composantes connexes de $G' \backslash S$ sont les mêmes que celles de $G \backslash S$, excepté $C_x$ et $C_y$ qui forment dans $G'\backslash S$ une unique composante connexe $C_x \cup C_y$. De plus, une composante connexe pleine induite par $S$ dans $G$ reste pleine dans $G'$. Comme $S$ est un séparateur minimal de $G$, d'après la proposition 2.14 $S$ induit deux composantes pleines $D$ et $E$ dans $G$. Si $D$ et $E$ sont toutes deux différentes de $C_x$ et $C_y$, alors $D$ et $E$ sont des composantes pleines associées à $S$ dans $G'$, donc $S$ est un séparateur minimal de $G'$. Supposons que $D = C_x$ et $E \neq C_y$. Alors $C_x \cup C_y$ est une composante connexe pleine associée à $S$ dans $G'$, car tout sommet de $S$ a un voisin dans $D = C_x$. De plus, $E$ reste une composante pleine associée à $S$ dans $G'$. Encore une fois, nous concluons que $S$ induit deux composantes connexes pleines dans $G'$, donc d'après la proposition2.14 $S$ est un séparateur minimal de $G'$. Enfin, si $D = C_x$ et $E = C_y$ ou symétriquement, alors $S$ est un $x, y$-séparateur minimal de $G$ d'après la proposition 2.14. Comme le seul $x, y$-séparateur minimal de $G$ est $S_{xy}$ il s'ensuit que $S = S_{xy}$.

Si $S$ ne sépare pas $x$ et $y$ dans $G$, les composantes connexes de $G'\backslash S$ sont les mêmes que celles de $G\backslash S$. Comme $S$ induit deux composantes pleines $C$ et $D$ dans $G$, ce seront aussi des composantes connexes de $G'\backslash S$ et de plus elles restent pleines par rapport à $S$ dans $G'$. Il s'ensuit que $S$ est un séparateur minimal de $G'$, d'après la proposition 2.14. $\diamond$

Nous en déduisons que pour chaque graphe $G_i$ produit par l'algorithme de reconnaissance, $G_i$ a au plus un séparateur minimal de plus que $G_{i+1}$, pour $0 \leq i \leq \overline{m}$. Comme $G_{\overline{m}}$ est une clique, il n'a pas de séparateurs. On en conclut :

**Corollaire 5.22** *Un graphe faiblement triangulé $G$ a au plus $\overline{m}$ séparateurs minimaux, où $\overline{m}$ est le nombre d'arêtes de son complémentaire $\overline{G}$.*

Nous devons signaler que ce résultat a été découvert indépendemment par Ton Kloks, qui ne l'a pas publié, mais qui est cité dans [Kra96].

Observons que ce procédé nous donne aussi le moyen de calculer les séparateurs minimaux d'un graphe faiblement triangulé $G_0$. Nous utilisons l'algorithme de reconnaissance qui produit itérativement les graphes $G_i$, où $G_{\overline{m}}$ est une clique et donc n'a pas de séparateurs. Il ne reste plus qu'à revenir en arrière et calculer les séparateurs minimaux de $G_i$ en fonction de ceux de $G_{i+1}$.

Ceci nous a donné l'idée de tenter le même type de procédé pour calculer les cliques maximales potentielles d'un graphe faiblement triangulé. Replaçons-nous dans le cas des graphes quelconques.

Nous considérons un graphe quelconque $G = (V, E)$ et deux sommets non adjacents $x$ et $y$ de $G$. Comme précédemment, posons $G' = (V, E \cup \{xy\})$ le graphe obtenu en rajoutant l'arête $xy$. Nous décrirons les cliques maximales potentielles de $G$ en fonction des séparateurs minimaux de $G$ et des cliques maximales potentielles de $G'$. Rappelons que d'après le théorème 3.17, $\Omega$ est une clique maximale potentielle de $G$ si et seulement si aucune composante $C_i$ de $G\backslash\Omega$ n'est pleine par rapport à $\Omega$ et, en complétant les ensembles $S_i$ de sommets de $\Omega$ ayant des voisins dans $C_i$, $\Omega$ devient une clique. Nous considérons toutes les positions possibles de $x$ et $y$ par rapport à $\Omega$ et aux composantes connexes de $G\backslash\Omega$. Nous verrons que, sauf pour des cas très particuliers, $\Omega$ reste une clique maximale potentielle dans $G'$.

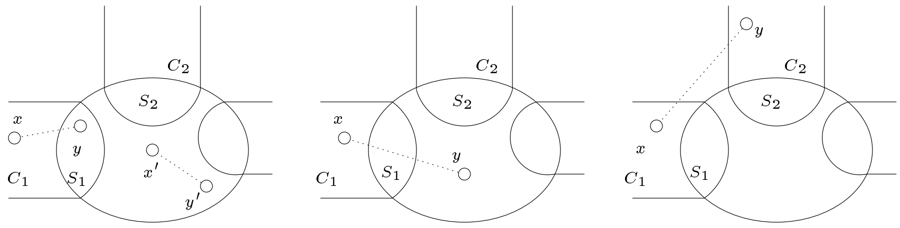


FIG. 5.9: Cliques maximales potentielles et ajout d'une arête

**Théorème 5.23** *Soient $G = (V, E)$ un graphe, $x$ et $y$ deux sommets non adjacents de $G$ et $G' = (V, E \cup \{xy\})$. Soient $\Omega$ une clique maximale potentielle de $G$, $C_1, \ldots, C_p$ les composantes connexes de $G \backslash \Omega$ et $S_i$ les ensembles de sommets de $\Omega$ ayant un voisin dans $C_i$.*

1. *Si $x, y \in \Omega$ ou $x, y \in C_1 \cup S_1$, alors $\Omega$ est une clique maximale potentielle de $G'$.*
2. *Si $x \in C_1$, $y \in \Omega \backslash S_1$ et $\Omega \neq S_1 \cup \{y\}$, alors $\Omega$ est une clique maximale potentielle de $G'$.*
3. *Si $x \in C_1$, $y \in C_2$ et $\Omega \neq S_1 \cup S_2$, alors $\Omega$ est une clique maximale potentielle de $G'$.*

*Preuve.* Les trois situations sont illustrées par la figure 5.9.

Rappelons que les $S_i$ sont strictement contenus dans $\Omega$ car $\Omega$ n'induit pas de composante connexe pleine dans $G$ et que $\Omega$ est une clique dans $G_{\{S_1,\ldots,S_p\}}$. La preuve se fait directement à partir du théorème 3.17. Il suffit de regarder les composantes connexes $C'_1, \ldots, C'_q$ de $G' \backslash \Omega$ et les ensembles $S'_i$, $1 \leq i \leq q$ de sommets de $\Omega$ ayant un voisin dans $C'_i$ dans le graphe $G'$. Nous allons procéder cas par cas.

Si $x, y \in \Omega$ ou $x, y \in S_1 \cup C_1$, les composantes connexes de $G' \backslash \Omega$ sont les mêmes que celles de $G \backslash \Omega$ et les ensembles $S'_i$ sont identiques aux $S_i$. Donc $\Omega$ n'induit pas de composante pleine dans $G'$, et $G'_{\{S'_1,\ldots,S'_q\}}[\Omega] = G_{\{S_1,\ldots,S_p\}}[\Omega]$ est une clique. Il s'ensuit que $\Omega$ satisfait dans le graphe $G'$ les conditions du théorème 3.17.

Si $x \in C_1$ et $y \in \Omega \backslash S_1$, alors $C'_i = C_i$ pour $1 \leq i \leq p$. Observons que $S'_1 = S_1 \cup \{y\}$ et $S'_i = S_i$ pour $i \geq 2$. Nous devons vérifier si $\Omega$ induit une composante pleine dans $G'$. Clairement les composantes $C'_2 = C_2, \ldots, C'_p = C_p$ ne sont pas pleines par rapport à $\Omega$ car $S'_i = S_i \subset \Omega$ pour $i \geq 2$. Si de plus $\Omega \neq S_1 \cup \{y\}$, nous en déduisons que $S'_1$ est strictement contenu dans $\Omega$ et donc $\Omega$ n'induit pas de composante connexe pleine dans $G'$. Il est évident que $\Omega$, qui est une clique dans $G_{\{S_1,\ldots,S_p\}}$, reste une clique dans $G'_{\{S'_1,\ldots,S'_q\}}$. D'après le théorème 3.17, $\Omega$ est une clique maximale potentielle de $G'$.

Si $x \in C_1$ et $y \in C_2$, les composantes connexes de $G' \backslash \Omega$ seront $C'_1 = C_1 \cup C_2$, $C'_2 = C_3, \ldots, C'_{p-1} = C_p$. Les ensembles $S'_i$ correspondants sont $S'_1 = S_1 \cup S_2$, $S'_2 = S_3, \ldots, S'_{p-1} = S_p$. Dans ce cas aussi, les ensembles $S'_2, \ldots, S'_{p-1}$ sont strictement contenus dans $\Omega$. Si $\Omega \neq S_1 \cup S_2$, nous avons aussi $S'_1 \subset \Omega$ et donc $\Omega$ n'induit pas de composante pleine dans $G'$. Clairement, $G'_{\{S'_1,\ldots,S'_q\}}[\Omega]$ est une clique, donc $\Omega$ est une clique maximale potentielle de $G'$ d'après le théorème 3.17. $\diamond$

Remarquons que les trois cas du théorème 5.23 couvrent tous les positionnements possibles de $x$ et $y$ par rapport à $\Omega$ et aux composantes connexes de $G\backslash\Omega$. On en déduit facilement :

**Corollaire 5.24** *Soit $\Omega$ une clique maximale potentielle de $G = (V, E)$. Soient $x$ et $y$ deux sommets non adjacents de $G$ et $G' = (V, E \cup \{xy\})$. Deux situations sont possibles :*

- *$\Omega$ s'écrit sous la forme $S_1 \cup \{x\}$, $S_1 \cup \{y\}$ ou $S_1 \cup S_2$, où $S_1$, $S_2$ sont des $x, y$-séparateurs minimaux de $G$.*
- *$\Omega$ est une clique maximale potentielle de $G'$.*

*Preuve.* La seule chose à prouver est que si $\Omega$ est de la forme $S_1 \cup \{y\}$ ou $S_1 \cup S_2$, alors $S_1$ et $S_2$ sont bien des $x, y$-séparateurs minimaux.

Si $\Omega = S_1 \cup \{y\}$, nous sommes dans le cas où $x \in C_1$ et $y \in \Omega\backslash S_1$. Clairement, $x$ et $y$ sont dans des composantes pleines différentes associées à $S_1$, donc $S_1$ est un $x, y$-séparateur minimal.

Si $\Omega = S_1 \cup S_2$, nous sommes dans le cas $x \in C_1$, $y \in C_2$. Notons que $S_1$ ne peut pas être contenu dans $S_2$, on aurait $\Omega = S_2$ ce qui est impossible puisque $\Omega$ n'induit pas de composante pleine. Soit $y'$ un sommet de $S_2\backslash S_1$. Comme $C_2$ est une composante pleine associée à $S_2$ et $y \in C_2$, il existe une chaîne reliant $y$ à $y'$, contenue dans $C_2 \cup \{y'\}$. Cette chaîne n'intersecte pas $S_1$, il s'ensuit que $y'$ et $y$ sont dans la même composante connexe de $G\backslash S_1$. On en déduit que $y$ est dans la composante de $G\backslash S_1$ qui intersecte $\Omega$, et d'après la propriété 3.13 cette composante associée à $S_1$ est pleine. Nous en concluons que $x$ et $y$ sont dans des composantes connexes distinctes de $G\backslash S_1$, pleines par rapport à $S_1$, donc $S_1$ est bien un $x, y$-séparateur minimal. De la même manière, $S_2$ est un $x, y$-séparateur minimal. $\diamond$

Revenons maintenant aux graphes faiblement triangulés. Si l'on veut calculer les cliques maximales potentielles d'un graphe faiblement triangulé $G_0$, nous pouvons utiliser l'algorithme de reconnaissance pour produire les graphes $G_1, G_2, \ldots, G_{\overline{m}}$ où $G_{i+1}$ est obtenu à partir de $G_i$ en ajoutant une arête entre les sommets d'une deux-paire de $G_i$. Nous pouvons calculer les séparateurs minimaux de chaque $G_i$, donc nous pouvons calculer les cliques maximales potentielles de $G_i$ en fonction de celles de $G_{i+1}$. Comme $G_{\overline{m}}$ est une clique, sa seule clique potentielle est $V(G_{\overline{m}})$. De cette manière, nous obtenons toutes cliques maximales potentielles de $G_0$.

Mais dans le cas particulier des graphes faiblement triangulés, nous pouvons affiner les résultats du théorème 5.23, en éliminant le troisième cas.

**Proposition 5.25** *Soit $G = (V, E)$ un graphe faiblement triangulé non complet et soit $\{x, y\}$ une deux-paire de $G$. Considérons une clique maximale potentielle $\Omega$ de $G$ telle que $x$ et $y$ soient dans des composantes connexes distinctes de $G\backslash\Omega$. Alors $\Omega$ est une clique maximale potentielle du graphe $G' = (V, E \cup \{xy\})$.*

*Preuve.* Nous utilisons le fait que si $\{x, y\}$ est une deux-paire de $G$, alors leurs voisins communs $S_{xy} = N(x) \cap N(y)$ forment l'unique $x, y$-séparateur minimal de $G$.

Soient $C_1$ et $C_2$ les composantes connexes de $G \backslash \Omega$ contenant $x$, respectivement $y$. Comme précédemment, nous notons par $S_1$, respectivement $S_2$ les ensembles de sommets de $\Omega$ ayant des voisins dans $C_1$, respectivement $C_2$. Nous voulons prouver que $S_1$ et $S_2$ contiennent $S_{xy}$ et que au moins l'un d'entre eux est égal à $S_{xy}$. Comme $(S_1, C_1)$ est un bloc de $S_1$ dans $G$ par le théorème 3.17, il s'ensuit que $S_1$ sépare $x$ et $y$ dans $G$. En particulier, $S_1$ contient $S_{xy}$. De façon symétrique, $S_2$ contient $S_{xy}$. Supposons maintenant qu'il existe un sommet $a \in S_1 \backslash S_{xy}$ et un sommet $b \in S_2 \backslash S_{xy}$. Comme $a$ a un voisin dans $C_1$, il est dans la composante connexe de $G \backslash S_{xy}$ contenant $x$. De même, $b$ est dans la composante connexe de $G \backslash S_{xy}$ contenant $y$. Nous en déduisons que $\Omega$ intersecte deux composantes connexes différentes de $G \backslash S_{xy}$. Mais $S_{xy}$ est un séparateur minimal contenu dans $\Omega$, et d'après la proposition 3.13, $\Omega$ doit être contenu dans un bloc de $S_{xy}$, contredisant le fait que $S_{xy}$ sépare $a$ et $b$. Donc au moins un des deux ensembles $S_1$ ou $S_2$ est égal à $S_{xy}$. Il s'ensuit que $S_1 \cup S_2$ est égal à $S_1$ ou à $S_2$. Comme $\Omega$ n'est pas un séparateur minimal de $G$, dans les deux cas $S_1 \cup S_2 \neq \Omega$. Par le troisième cas du théorème 5.23, $\Omega$ est une clique maximale potentielle de $G$. $\diamond$

Remarquons que la proposition 5.25 reste vraie même si le graphe $G$ n'est pas faiblement triangulé, il suffit seulement qu'il possède une deux-paire $\{x, y\}$.

Nous pouvons reconsidérer le corollaire 5.24 pour le cas particulier des graphes faiblement triangulés, voire des graphes possédant une deux-paire :

**Corollaire 5.26** *Soit $\Omega$ une clique maximale potentielle du graphe faiblement triangulé $G = (V, E)$. Soient $x$ et $y$ deux sommets formant une deux-paire de $G$ et $G' = (V, E \cup \{xy\})$. Deux situations sont possibles :*

- *$\Omega$ s'écrit sous la forme $S_{xy} \cup \{x\}$ ou $S_{xy} \cup \{y\}$, avec $S_{xy} = N(x) \cap N(y)$.*
- *$\Omega$ est une clique maximale potentielle de $G'$.*

**Corollaire 5.27** *Un graphe faiblement triangulé $G$ a au plus $2\overline{m} + 1$ cliques maximales potentielles.*

*Preuve.* Il suffit de considérer les graphes $G = G_0, G_1, \ldots, G_{\overline{m}}$ obtenus par ajout successif d'une arête entre les sommets d'une deux-paire et d'observer que $G_i$ a au plus deux cliques maximales potentielles de plus que $G_{i+1}$. Comme $G_{\overline{m}}$ est une clique, sa seule clique maximale potentielle est la totalité de ses sommets. $\diamond$

On en déduit directement :

**Corollaire 5.28** *Les cliques maximales potentielles d'un graphe faiblement triangulé peuvent être énumérées en temps polynomial.*

*Preuve.* L'algorithme 5.2 calcule en temps polynomial les cliques potentielles maximales du graphe faiblement triangulé $G_0$.

Pour calculer les deux-paires $x_i, y_i$ on utilise l'algorithme de reconnaissance des graphes faiblement triangulés de Spinrad et Sritharan [SS95] qui fonctionne en temps $\mathcal{O}(n^2\overline{m})$. On peut calculer en même temps et les ensembles $S_{x_i y_i}$, et enfin les $2\overline{m}$

**Algorithme 5.2 : Cliques maximales potentielles des graphes faiblement triangulés**

`Entrée` : le graphe faiblement triangulé $G_0$

`Sortie` : ses séparateurs minimaux et ses cliques maximales potentielles

`pour` $i = 1$ à $\overline{m}$
  calculer une deux-paire $x_i, y_i$ de $G_{i-1}$
  $S_{x_iy_i} \leftarrow N_{G_{i-1}}(x_i) \cap N_{G_{i-1}}(y_i)$
  $G_i \leftarrow (V, E(G_{i-1}) \cup \{x_iy_i\})$

  vérifier si $S_{x_iy_i}$ est un séparateur minimal de $G_0$
  vérifier si $S_{x_iy_i} \cup \{x_i\}$ et $S_{x_iy_i} \cup \{y_i\}$ sont des cliques potentielles de $G_0$

ensembles $S_{x_iy_i} \cup \{x_i\}$ et $S_{x_iy_i} \cup \{y_i\}$ qui pourraient être des cliques maximales potentielles. Enfin, étant donné un ensemble de sommets $K$ d'un graphe quelconque $G$, on peut vérifier en temps $\mathcal{O}(n^3)$ si $K$ est une clique potentielle de $G$ (cf. corollaire 3.18). Ceci fait que l'on peut obtenir les cliques maximales potentielles d'un graphe faiblement triangulé en temps $\mathcal{O}(n^3\overline{m})$. $\diamond$

## 5.4 Conclusion

Nous avons vu dans ce chapitre que, pour toutes les classes de graphes pour lesquelles nous savions calculer la largeur arborescente, nous savons en fait énumérer en temps polynomial les cliques maximales potentielles. De plus, tous les algorithmes déjà cités calculent de manière implicite toutes les cliques potentielles du graphe en entrée. La première conclusion est que la notion de clique potentielle unifie en quelque sorte les algorithmes respectifs.

La classe des graphes faiblement triangulés est à notre connaissance la seule classe naturelle avec un nombre polynomial de séparateurs minimaux et pour laquelle le calcul de la largeur arborescente et de la triangulation minimale étaient encore des problèmes ouverts. Pour ces graphes, nous avons pu aussi énumérer toutes les cliques maximales potentielles, et par conséquent calculer les deux paramètres de triangulation. Il est à noter que, si l'on sait énumérer les cliques maximales potentielles, on ne sait rien dire sur leur structure.

Nous avions montré au chapitre 4 que les cliques maximales potentielles suffisent pour calculer la largeur arborescente et la complétion minimale d'un graphe. Les résultats de ce chapitre nous permettent de penser que, pour les graphes avec "peu" de séparateurs, le nombre de cliques maximales potentielles est polynomial est qu'il est possible d'énumérer en temps polynomial toutes ces cliques potentielles. En tout cas, les classes de graphes pour lesquelles nous savons calculer les paramètres de triangulation coïncident avec les classes de graphes pour lesquelles l'on sait énumérer les cliques maximales potentielles...

Malgré les caractérisations des cliques maximales potentielles données dans le chapitre 3 par les théorèmes 3.11 et 3.12 ou encore 3.17, nous avons peu d'informations susceptibles de nous aider à énumérer les cliques maximales potentielles d'un graphe quelconque, comme il a été fait pour les séparateurs minimaux. Nous ignorons même si le nombre de cliques potentielles des graphes quelconques dépend polynomialement du nombre de séparateurs minimaux du graphe.

Toutefois, une petite ouverture nous est suggérée par le théorème 5.23 et son corollaire 5.24. D'après ces assertions, si nous disposons des séparateurs minimaux d'un graphes $G = (V, E)$ et des séparateurs minimaux, ainsi que des cliques maximales potentielles d'un graphe $G' = (V, E \cup \{xy\})$ obtenu en rajoutant une arête à $G$, alors nous pouvons énumérer les cliques maximales potentielles de $G$. Si l'on pouvait, comme nous l'avons fait pour les graphes faiblement triangulés, calculer une suite de graphes $G_0 = G, G_1, \ldots, G_{\overline{m}}$ telle que $G_{i+1}$ soit obtenu à partir de $G_i$ par rajout d'une arête, $G_{\overline{m}}$ soit une clique et le nombre de séparateurs minimaux de chaque $G_i$ dépende de façon polynomiale de $n$ et du nombre de séparateurs minimaux de $G$, alors nous saurions calculer les cliques maximales potentielles de $G$ en un temps qui dépend polynomialement de $n$ et de $|\Delta_G|$.

Plus formellement, nous proposons la conjecture suivante :

**Conjecture 5.29** *Il existe un polynôme $P$ tel que pour tout graphe $G$, il existe une suite de graphes $G_0 = G, G_1, \ldots, G_{\overline{m}}$, où $G_{\overline{m}}$ est une clique, pour tout $i$, $G_{i+1}$ est obtenu en rajoutant une arête à $G_i$ et $|\Delta_{G_i}| \leq P(n, |\Delta_G|)$.*

A priori, en rajoutant une arête à un graphe avec $r$ séparateurs, nous pouvons obtenir un graphe avec $\Theta(r^2)$ séparateurs, comme on peut s'en convaincre en prenant comme graphe $G$ une chaîne $P_n$ à $n$ sommets et en la transformant en un cycle $C_n$. Néanmoins, en regardant le problème de plus près, nous n'avons pas pu prouver ou infirmer cette petite conjecture.

# Chapitre 6

# Conclusion et perspectives

Tout le travail que nous venons de présenter porte sur les triangulations minimales des graphes, dans le but de calculer la largeur arborescente et la complétion minimale des graphes ayant un nombre polynomial de séparateurs.

Nous avons constaté que les séparateurs minimaux ne suffisent pas pour caractériser de manière algorithmique les triangulations minimales d'un graphe. Pour calculer la largeur arborescente d'un graphe $G$, il fallait maîtriser les cliques qui apparaissent dans les triangulations minimales de $G$. Pour cela nous avons d'abord regroupé les séparateurs minimaux qui bordent ces cliques, en introduisant la notion de famille maximale de séparateurs voisins. Nous avons constaté ensuite que nous pouvions nous affranchir des séparateurs minimaux et regarder directement les cliques maximales potentielles, c'est-à-dire les ensembles de sommets qui forment des cliques maximales dans des triangulations minimales de $G$. Cette dernière notion est plus intuitive et plus facile à manipuler. Nous avons prouvé que les cliques maximales potentielles ne sont pas de simples objets combinatoires, car ils peuvent être identifiés localement, dans le graphe $G$, et il existe un algorithme polynomial qui reconnaît si un ensemble de sommets $K$ est une clique maximale potentielle.

Nous avons montré que la connaissance de ces objets suffit pour calculer la largeur arborescente et la complétion minimale d'un graphe. D'un autre côté, nous avons donné des algorithmes d'énumération des cliques maximales potentielles pour toutes les classes de graphes pour lesquelles il existait des algorithmes polynomiaux calculant la largeur arborescente et la complétion minimale. Pour certains graphes nous avons une bonne connaissance de la structure des séparateurs minimaux, qui nous a permis de caractériser les séparateurs voisins et par conséquent d'énumérer les cliques potentielles. Pour d'autres classes, notamment les graphes faiblement triangulés, nous n'avons pas beaucoup d'information sur la structure des séparateurs voisins. Toutefois, nous avons pu énumérer directement les cliques maximales potentielles de ces graphes et nous avons prouvé ainsi que le calcul de la largeur arborescente et de la complétion minimale est polynomial pour les graphes faiblement triangulés.

En résumé, nous avons constaté que les graphes pour lesquels nous savons calculer les deux paramètres de triangulation sont exactement les graphes pour lesquels nous pouvons calculer les cliques maximales potentielles. Notre conviction est que la notion essentielle pour le calcul de la largeur arborescente et de la complétion minimale d'un graphe est la notion de clique maximale potentielle, et non celle de séparateur minimal. Rappelons encore une fois que la notion de clique maximale potentielle coïncide avec la notion de regroupement local de séparateurs minimaux, ce que nous avons appelé “séparateurs voisins”.

Hélas, le grand problème ouvert qui nous a inspiré ces travaux, la conjecture ESA'93 qui dit que le calcul de la largeur arborescente et de la complétion minimale devrait être polynomial pour les graphes avec “peu” de séparateurs, reste sans réponse. Pour toutes les classes de graphes que nous avons regardées, nous avons pu prouver que le nombre de cliques maximales potentielles est polynomial en le nombre de séparateurs minimaux, nous avons même donné des algorithmes pour les énumérer. Ceci reste vrai même pour des classes de graphes qui n'ont pas un nombre polynomial de séparateurs comme les graphes AT-free ou les graphes planaires. Néanmoins, nous n'avons pas pu exhiber un algorithme polynomial qui énumère toutes les cliques maximales potentielles d'un graphe quelconque, comme cela a été fait par exemple pour les séparateurs minimaux.

Poursuivre les travaux dans cette direction est la voie ouverte la plus naturelle que laisse cette thèse. Rappelons la conjecture qui clôt le chapitre précédent. Elle affirme que, en partant de n'importe quel graphe $G$, on pourrait lui rajouter successivement des arêtes jusqu'à ce qu'il devienne une clique et de sorte que le nombre de séparateurs minimaux des graphes intermédiaires soit polynomial par rapport au nombre des séparateurs minimaux de $G$. Elle nous semble un bon point de départ pour résoudre la conjecture ESA'93.

Un autre problème ouvert intéressant - et entre autres très frustrant - est le calcul de la largeur arborescente des graphes planaires. Seymour et Thomas [ST94] ont prouvé que le branchwidth, un paramètre qui ne diffère que d'une constante multiplicative de la largeur arborescente, est calculable en temps polynomial pour la classe des graphes planaires. Les graphes planaires ont en général un nombre exponentiel de séparateurs minimaux. Mais nous avons pu caractériser - et ces résultats ne se trouvent pas dans ce document - les cliques maximales potentielles des graphes planaires, qui sont une fois de plus simples et en nombre polynomial par rapport au nombre de séparateurs minimaux. Ne pourrait-on pas définir une “famille essentielle” de cliques maximales potentielles, qui soit calculable en temps polynomial et qui suffise pour le calcul de la largeur arborescente ? En tout cas, le calcul de la largeur arborescente des graphes planaires semble une direction très intéressante à suivre.

Notons que nous ne savons pas à ce jour si l'approximation de la largeur arborescente à une constante multiplicative près est un problème polynomial ou NP-difficile. Pour aller donc plus loin avec les graphes ayant beaucoup de séparateurs, se pose la question de savoir si peu de séparateurs et peu de cliques maximales potentielles pourraient suffire pour calculer ou approximer la largeur arborescente et la complétion minimale de certaines classes.

# Bibliographie


[ACP87] 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.

[ACPS93] S. Arnborg, B. Courcelle, A. Proskurowski, and D. Seese. An algebraic theory of graph reduction. *J. of ACM*, 40 :1134–1164, 1993.

[AP86] S. Arnborg and A. Proskurowski. Characterization and recognition of partial 3-trees. *SIAM J. on Algebraic and Discrete Methods*, 7 :305–314, 1986.

[AP89] S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems restricted to partial $k$-trees. *Discrete Applied Mathematics*, 23 :11–24, 1989.

[BB96] A. Berry and J.P. Bordat. Using minimal separation to compute minimal orderings. Research report, LIRMM, Montpellier, 1996. Submitted to Discrete Applied Mathematics.

[BB98] A. Berry and J.P. Bordat. Separability generalizes Dirac's theorem. *Discrete Applied Mathematics*, 84(1-3) :43–53, 1998.

[BdF96] 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.

[BDK97a] H.J. Broersma, E. Dahlhaus, and T. Kloks. Algorithms for the treewidth and minimum fill-in of HHD-free graphs. In *Workshop on Graphs (WG'97)*, volume 1335 of *Lecture Notes in Computer Science*, pages 109–117. Springer-Verlag, 1997.

[BDK97b] H.J. Broersma, E. Dahlhaus, and T. Kloks. A linear time algorithm for minimum fill in and treewidth for distance hereditary raphs. In *Scientific program 5th Twente Workshop on Graphs and Combinatorial Optimization*, pages 48–50, 1997.

[Ber95] A. Berry. Treillis de Galois des séparateurs minimaux d'un graphe non-orienté. Master's thesis, Université Montpellier II, 1995.

[BKK93] H.L. Bodlaender, T. Kloks, and D. Kratsch. Treewidth and pathwidth of permutation graphs. In *Proceedings of the 20th International Colloquium* *on Automata, Languages and Programming (ICALP'93)*, volume 700 of *Lecture Notes in Computer Science*, pages 114–125. Springer-Verlag, 1993.

[BKK95] H. Bodlaender, T. Kloks, and D. Kratsch. Treewidth and pathwidth of permutation graphs. *SIAM J. Discrete Math.*, 8(4) :606–616, 1995.

[BKKM98] H. Broersma, T. Kloks, D. Kratsch, and H. Müller. A generalization of AT-free graphs and a generic algorithm for solving triangulation problems. In *Workshop on Graphs WG'98*, Lecture Notes in Computer Science. Springer-Verlag, 1998.

[Bod93] H. Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11 :1–23, 1993.

[Bod96] H. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *Siam J. Computing*, 25 :1305–1317, 1996.

[Bod97] 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.

[BP93] J. R. S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. In A. George, J. R. Gilbert, and J. H. U. Liu, editors, *Graph Theory and Sparse Matrix Computations*, pages 1–29. Springer, 1993.

[BT95] H. Bodlaender and D.M. Thilikos. Treewidth and small separators for graphs with small chordality. Technical Report UU-CS-1995-02, Utrecht University, 1995. A paraître dans Discrete Applied Mathematics.

[BT98a] 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.

[BT98b] V. Bouchitté and I. Todinca. Treewidth and minimum fill-in of weakly triangulated graphs. Research Report RR98-40, LIP-ENS, 1998. A paraître dans Proceedings of STACS'99.

[Cha96] 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.

[CM93] B. Courcelle and M. Moshbah. Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science*, 109 :49–82, 1993.

[Cou92] B. Courcelle. The monadic second-order logic of graphs III : Treewidth, forbidden minors and complexity issues. *Informatique Théorique*, 26 :257–286, 1992.

[Dir61] G.A. Dirac. On rigid circuit graphs. *Abh. Math. Sem. Univ. Hamburg*, 21 :71–76, 1961.

[ES93] 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.

[Gav74] F. Gavril. The intersection graphs of a path in a tree are exactly the chordal graphs. *Journ. Comb. Theory*, 16 :47–56, 1974.

[Gol80] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

[Hag97] 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.

[Hay85] R. Hayward. Weakly triangulated graphs. *J. Combin. Theory ser. B*, 39 :200–208, 1985.

[HHM89] R. Hayward, C. Hoàng, and F. Maffray. Optimizing weakly triangulated graphs. *Graphs Combin.*, 5 :339–349, 1989.

[HL89] C.W. Ho and R.C.T. Lee. Counting clique trees and computing perfect elimination schemes in parallel. *Inform. Process. Letters*, 31 :61–68, 1989.

[HM94] M. Habib and R. Möhring. Treewidth of cocomparability graphs and a new order-theoretic parameter. *ORDER*, 1 :47–60, 1994.

[KBMK93] 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.

[KBMK94] 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.

[KK94] T. Kloks and D. Kratsch. Finding all minimal separators of a graph. In *Proceedings 11th Annual Symposium on Theoretical Aspects of Computer Science (STACS'94)*, volume 775 of *Lecture Notes in Computer Science*, pages 759–768. Springer-Verlag, 1994.

[KK95] T. Kloks and D. Kratsch. Treewidth of chordal bipartite graphs. *J. Algorithms*, 19(2) :266–281, 1995.

[KK98] T. Kloks and D. Kratsch. Listing all minimal separators of a graph. *SIAM J. Comput.*, 27(3) :605–613, 1998.

[KKM95] 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.

[KKM97] T. Kloks, D. Kratsch, and H. Müller. Asteroidal sets in graphs. In *Workshop on Graphs (WG'97)*, volume 1335 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.

[KKM98] T. Kloks, D. Kratsch, and H. Müller, 1998. Private communication.

[KKS97] 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.

[KKW98] T. Kloks, D. Kratsch, and C.K. Wong. Minimum fill-in of circle and circular-arc graphs. *J. Algorithms*, 28(2) :272–289, 1998.

[Klo94] T. Kloks. *Treewidth - Computations and Approximations*, volume 842. Springer-Verlag, 1994.

[Klo96] T. Kloks. Treewidth of circle graphs. *Intern. J. Found. Comput. Sci.*, 7 :111–120, 1996.

[Kra96] D. Kratsch. *The structure of Graphs and the Design of Efficient Algorithms*. Habilitation thesis, F. Schiller Universität, Jena, 1996.

[LB62] C.G. Lekkerkerker and J. Ch. Boland. Representation of a finite graph by a set of intervals on the real line. *Fundamenta Mathematicae*, 51 :45–64, 1962.

[Lun90] M. Lundquist. *Zero Patterns, Chordal Graphs and Matrix Completion*. PhD thesis, Clemson University, 1990.

[Möh96] R.H. Möhring. Triangulating graphs without asteroidal triples. *Discrete Applied Mathematics*, 64 :281–287, 1996.

[Par96] A. Parra. *Structural and Algorithmic Aspects of Chordal Graph Embeddings*. PhD thesis, Technische Universität Berlin, 1996.

[Pau98] Ch. Paul. *Parcours en Largeur Lexicographique : Un Algorithme de Partitionnement. Application aux Graphes et Généralisation*. PhD thesis, Université Montpellier II, 1998.

[PS97] A. Parra and P. Scheffler. Characterizations and algorithmic applications of chordal graph embeddings. *Discrete Appl. Math.*, 79(1-3) :171–188, 1997.

[Ros70] D.J. Rose. Triangulating graphs and the elimination process. *J. Math Anal Appl.*, 32 :597–609, 1970.

[Ros74] D.J. Rose. On simple characterization of $k$-trees. *Discrete Mathematics*, 7 :317–322, 1974.

[RS84] N. Robertson and P. Seymour. Graphs minors. III. Planar tree-width. *J. of Combinatorial Theory Series B*, 36 :49–64, 1984.

[RS86] N. Robertson and P. Seymour. Graphs minors. II. Algorithmic aspects of tree-width. *J. of Algorithms*, 7 :309–322, 1986.

[RTL76] D.J. Rose, R.E. Tarjan, and G.D. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5 :266–283, 1976.

[San96] D.P. Sanders. On linear recognition of tree-width at most four. *Siam J. Disc. Meth.*, 9(1) :101–117, 1996.

[SL97] H. Shen and W. Liang. Efficient enumeration of all minimal separators in a graph. *Theoretical Computer Science*, 180 :169–180, 1997.

[Spi94] J. Spinrad. Recognition of circle graphs. *J. Algorithms*, 16 :264–282, 1994.

[SS95] J. Spinrad and R. Sritharan. Algorithms for weakly triangulated graphs. *Discrete Applied Mathematics*, 59 :181–191, 1995.

[SSR94] R. Sundaram, K. Sher Singh, and C. Pandu Rangan. Treewidth of circular-arc graphs. *SIAM J. Discrete Math.*, 7 :647–655, 1994.

[ST94] P.D. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2) :217–241, 1994.

[Wal78] J.R. Walter. Representations of chordal graphs as subtrees of a tree. *Journal of Graph Theory*, 2 :256–267, 1978.

[Yan81] M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 2 :77–79, 1981.

# Index


$(S, C)$, 23
$C_G^a(S)$, 17
$G_\Gamma$, 24
$N(x)$, 11
$S_{xy}$, 80
$\Delta_G$, 21, 80
$\Delta_G(K)$, 33
$\|$, 23
$\mathcal{C}_G^*(S)$, 17
$\mathcal{C}_G(S)$, 17
cm, 17
$\omega$, 11
$\overline{G}$, 11
$\sharp$, 23
tw, 15
$m$, 11
$n$, 11
$p$, 44, 55

arbre, 12
    de cliques, 13, 20

bloc, 23, 34, 37
    plein, 23

chaîne, 12
    induite, 12
    sans corde, 12
clique, 11
    maximale, 11
clique maximale potentielle, 35, 54, 62
    énumération, 69, 76, 78, 84
    caractérisation, 37, 38, 42
    famille complète, 48
    reconnaissance, 44
complétion minimale, 17
    calcul, 62
composante connexe, 12
    associée, induite, 17
    pleine, 17, 23, 42
corde, 12
corde de contrôle, 70
croisement, 23
cycle, 12, 79
    induit, 12
    sans corde, 12

décomposition arborescente, 15
deux-paire, 79
Dirac, 19

ensemble astéroïde, 69

famille maximale de séparateurs voisins, 35

graphe, 11
    complémentaire, 11, 79
    complet, 11
    connexe, 12
    d'intersection, 12, 70
    d'intervalles, 12, 69
    d'intervalles circulaires, 13, 70
    de cordes, 12, 70
    des séparateurs, 23
    faiblement triangulé, 79
    HHD-free, 79
    sans triplet astéroïde, 66
    triangulé, 13

hérédité, 13

largeur arborescente, 15, 16
    calcul, 54, 62
    linéaire, 69

modèle d'intersection, 12, 70

nombre astéroïde, 69

ordre, 12
ordre d'élimination simplicielle, 14

parallélisme, 23, 24
partie entre, 33, 34, 38
pathwidth, 69
point de contrôle, 70
programmation dynamique, 57
propriété d'intersection des cliques, 13

région élémentaire, 74

séparateur, 17
    $a, b$-séparateur, 17
    $a, b$-séparateur minimal, 17
    minimal, 17
séparateur minimal, 17, 19, 28
    énumération, 21
    le plus proche, 20, 22
sous-graphe, 11
    induit, 11

treewidth, 15
treillis, 12, 22
triangulation, 15
    minimale, 15, 17, 28, 78
    planaire, 78
triplet astéroïde, 66

voisinage, 11