



HAL
open science

Décomposition algorithmique des graphes

Frédéric Mazoit

► **To cite this version:**

Frédéric Mazoit. Décomposition algorithmique des graphes. Mathématiques [math]. Ecole normale supérieure de lyon - ENS LYON, 2004. Français. NNT : . tel-00148807

HAL Id: tel-00148807

<https://theses.hal.science/tel-00148807v1>

Submitted on 23 May 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : 302

N° attribué par la bibliothèque : 04ENSL302

ÉCOLE NORMALE SUPÉRIEURE DE LYON
Laboratoire de l'Informatique du Parallélisme

THÈSE

Soutenue publiquement le 16 décembre 2004
à l'École normale supérieure de Lyon pour obtenir le diplôme de doctorat
(spécialité : Informatique)
Au titre de l'école doctorale : mathématiques et informatique fondamentales

Frédéric Mazoit

Décompositions algorithmiques des graphes

Après avis de :

Monsieur Pierre Fraigniaud,	Membre/Rapporteur
Monsieur André Raspaud,	Membre/Rapporteur

Devant la commission d'examen formée de :

Monsieur Vincent Bouchitté,	Membre/Directeur de thèse
Monsieur Victor Chepoi,	Membre
Monsieur Pierre Fraigniaud,	Membre/Rapporteur
Monsieur Jean-Claude König,	Membre
Monsieur Jacques Mazoyer,	Membre
Monsieur André Raspaud,	Membre/Rapporteur
Monsieur Ioan Todinca,	Invité

À Céline,
la plus mieux bien des femmes,
tout simplement,

Remerciements

Je remercie Vincent Bouchitté pour m'avoir ouvert le monde que constituent les décompositions arborescentes et les décompositions en branches. Sans lui, cette thèse n'aurait jamais vu le jour.

Je remercie messieurs Pierre Fraigniaud et André Raspaud pour avoir accepté d'être les rapporteurs de ma thèse. Mon souci de rigueur et mes envolées formelles ne leur ont pas facilité la tâche. Je remercie messieurs Victor Chepoi et Jean-Claude König pour avoir bien voulu faire partie de mon jury de thèse et m'avoir consacré une journée en cette période chargée de fin d'année.

Je remercie Ioan Todinca qui a été un interlocuteur privilégié tout au long de ces années. Les discussions que nous avons eues m'ont guidé dans mes recherches et m'ont permis de préciser bon nombre d'idées encore floues. Son expérience et sa disponibilité durant cette phase difficile qu'est la rédaction du manuscrit m'ont été d'une aide précieuse.

Je tiens à remercier Jacques Mazoyer pour le travail administratif qu'il a fait. Merci aussi d'avoir été le « personnel de rang A de l'établissement d'origine » de mon jury de thèse.

Je ne remercie pas Laurent Lyaudet. Après avoir subi mes remarques sur son mémoire de DEA, il a eu l'idée saugrenue d'inverser les rôles et de faire une lecture minutieuse de mon manuscrit juste pour avoir le plaisir de me retourner mes critiques. De plus, il devrait être fouetté pour avoir demandé à sa gentille maman de relire tout ce charabia – celle-ci mérite la médaille du courage pour s'être plongée dans cet océan de graphes. Franchement Laurent, tu n'as pas honte ? Ceci dit, j'espère bien que nous travaillerons à nouveau ensemble d'ici peu.

Je remercie Benoît Kloeckner qui, en acceptant de relire le chapitre 8 a bien voulu jouer au « matheux de service ». Merci aussi pour toutes tes remarques typographiques.

Je remercie aussi Emmanuel Jeandel pour les discussions « graphiques » que nous avons eues.

Je remercie aussi David Lemke de Network Computing Devices, Heather Rose et Donald R. Woods de Sun Microsystems, Inc. et Yumi Hotta et Takeshi Obata de Jump Comics sans qui mes pauses n'auraient pas été les mêmes.

Je remercie les membres de l'équipe MC2 du LIP qui m'ont gentiment accueilli durant ces années. La bonne humeur ambiante m'a soutenu durant les périodes où mes théorèmes s'obstinaient à m'échapper et où j'en arrivais à me dire que je ferais mieux de changer de voie.

Et par dessus tout, je tiens à remercier Céline. Merci de m'avoir rappelé qu'une phrase comporte généralement un sujet et un verbe. Merci d'avoir relu minutieusement mes démonstrations. Tes « là, je ne comprend pas » m'ont évité de donner un bon nombre de théorèmes faux et sans tes remarques, cette thèse aurait été complètement incompréhensible. Je me joins à Marianne Delorme et Jacques Mazoyer pour dire qu'elle devrait nous être attribuée à tous les deux.

Sommaire

1	Introduction	1
1.1	Résumé de la thèse	3
2	Préliminaires	7
2.1	Terminologie	7
2.2	Séparateurs minimaux et frontières	10
2.2.1	Séparateurs minimaux	10
2.2.2	Frontières	13
2.3	Graphes et hyper-graphes triangulés	15
2.3.1	Séparateurs minimaux des hyper-graphes triangulés	16
2.3.2	Théorèmes de caractérisation	17
2.3.3	Cliques maximales des hyper-graphes triangulés	20
3	Énumération des séparateurs minimaux	23
3.1	Le treillis des a,b -séparateurs minimaux	24
3.2	Comment produire des séparateurs minimaux	25
3.3	L'algorithme d'énumération de Berry et col.	28
3.4	Un second algorithme	30
4	Diviser pour régner	35
4.1	Décompositions arborescentes	35
4.2	Décompositions en branches	40
4.3	Matriochkas et arbres de matriochkas	43
4.3.1	Matriochkas	44
4.3.2	Arbres de matriochkas	45
4.3.3	Extensions d'arbres de matriochkas	47
4.4	Quelques liens entre ces deux décompositions	50
5	Triangulations	57
5.1	Triangulations serrées	57
5.2	Familles complètes de blocs d'un hyper-graphe	60
6	Calcul de décompositions	65
6.1	Largeur arborescente	65
6.2	Largeur de branches	66
6.2.1	Hyper-graphes et triangulations	67
6.2.2	Largeur de branches des cliques et hyper-cliques	71
6.2.3	Profil	75
6.2.4	Le théorème de décomposition serrée	80

7	Application à quelques classes de graphes	85
7.1	Graphes de nombre astéroïde borné	85
7.2	Graphes d'intervalles circulaires	87
7.3	Autres classes de graphes	93
8	Graphes planaires	97
8.1	Séparateurs, séparations et graphe intermédiaire	99
8.1.1	Topologie des graphes et hyper-graphes planaires . . .	99
8.1.2	Courbes de Jordan et séparations	102
8.1.3	Homotopie et graphe intermédiaire	103
8.1.4	Séparateurs minimaux des graphes planaires	110
8.2	Énumération des séparateurs minimaux	115
8.3	Représentations de décompositions	119
8.4	Dualité et largeur arborescente	122
8.4.1	Définition et théorème de dualité	122
8.4.2	Dualité et graphe intermédiaire	125
8.4.3	Deux théorèmes de dualité	126
9	Conclusion	131

Notations

B_S , 30	θ , 35
$C_G^*(S)$, 10	$\text{tw}(G)$, 38
$C_G^a(S)$, 10	xy , 8
$C_G(S)$, 10	
$E(G)$, 8	
$G[E']$, 9	
$G[V']$, 9	
G_Γ , 58	
G_Ω , 67	
G/E , 41	
$N_G(x)$, 9	
$S_{a \rightarrow x}$, 26	
T , 10	
$T_{\mathcal{C}}$, 46	
T_θ , 54	
$V(G)$, 8	
$V(\lambda)$, 88	
$V_{G_\Sigma}(\mu)$, 102	
$\Delta_G(\Omega)$, 10	
$\Gamma_G(\Omega)$, 21	
$\Gamma_{\mathcal{F}}$, 60	
Σ_{G_Σ} , 102	
$\beta(G)$, 8	
$*T$, 10	
$\text{bw}(G)$, 40	
\mathcal{C} , 45	
\mathcal{C}_T , 46	
$\mathcal{D}(G)$, 87	
\mathcal{D}_T , 45	
$\mathcal{P}_G(\mathcal{S})$, 21	
\mathcal{S}_a , 30	
$\mathcal{S}_{a,B}$, 30	
$\mathcal{S}_{a,B}(S, O)$, 30	
$\mathcal{S}_{a,B}^x(S, O)$, 30	
$\gamma(G)$, 8	
$\partial(X)$, 13	
\preceq_a , 24	
τ , 40	
$\tau_{\mathcal{C}}$, 46	

Pour résoudre un problème, une démarche usuelle consiste à s'intéresser à des cas particuliers plus simples puis à ramener le problème général à ces cas particuliers. Dans le domaine des graphes, pour appliquer cette démarche, nous pouvons essayer de transformer des problèmes complexes sur des graphes en des problèmes sur des arbres. C'est ainsi qu'un certain nombre de décompositions de graphes en des structures arborescentes ont vu le jour. Nous étudierons dans cette thèse deux types de décompositions : les décompositions arborescentes et les décompositions en branches.

Certains graphes possèdent naturellement une structure arborescente : c'est le cas des graphes triangulés. Ceux-ci apparaissent au cours de l'étude de la conjecture des graphes parfaits dans les travaux de Dirac [Dir61] sous le nom de « rigid circuit graphs » et dans ceux de Hajós (1958). Ces graphes contenant les graphes complets, il est toujours possible de plonger un graphe G dans un graphe triangulé H . De cette façon, la structure arborescente de H se transpose à G . Cependant, lors d'un tel plongement, nous perdons de l'information sur la structure de G ; il est donc nécessaire de trouver un plongement ayant de « bonnes » propriétés. C'est dans ce cadre que, dans les années 70, sont apparus les k -arbres et les k -arbres partiels dans les travaux de Rose [Ros74] et de Dirac.

En 1960, Kruskal [Kru60] démontre que dans une famille infinie d'arbres, au moins l'un des arbres est mineur¹ d'un autre. Dans les années 80, Robertson et Seymour s'intéressent à la généralisation de ce théorème à des familles infinies de graphes et nomment ce problème la conjecture de Wagner en faisant référence à leur interprétation de [Wag37]. Pour démontrer ce théorème, ils essaient de se ramener au résultat de Kruskal et réintroduisent [RS84, RS86] à cet effet les décompositions arborescentes qui avaient déjà été définies en 1976 par Halin [Hal76] sous un autre nom. Pour pouvoir appliquer le résultat de Kruskal, ils ont besoin de mesurer à quel point un graphe ressemble à un arbre et utilisent pour cela un paramètre associé aux décompositions arborescentes : la largeur arborescente notée $tw(G)$. Lors de cette série d'articles, ils obtiennent de nombreux résultats importants, entre autres une généralisation du théorème de Kuratowski : pour toute classe de graphes \mathcal{C} close par minoration, il existe une famille finie $\mathcal{F}_{\mathcal{C}}$ de graphes telle que \mathcal{C} est exactement l'ensemble des graphes n'admettant aucun graphe de $\mathcal{F}_{\mathcal{C}}$ comme mineur. La famille $\mathcal{F}_{\mathcal{C}}$ est une famille d'obstructions pour la famille \mathcal{C} .

¹Un graphe est un mineur d'un graphe G s'il est obtenu en retirant des arêtes ou des sommets ou en contractant des arêtes de G .

Contrairement aux travaux de Halin, passés inaperçus, ceux de Robertson et Seymour attirent l'attention. Arnborg, Corneil et Proskurowski [ACP87] montrent que le problème de décision « le graphe G est de largeur arborescente k » est NP-complet. Cependant, ils montrent que si le paramètre k est fixé, ce problème devient polynomial et ils fournissent un algorithme de complexité $O(n^{k+2})$ pour le résoudre et construire le cas échéant une décomposition de largeur arborescente au plus k . Robertson et Seymour [RS95] améliorent le résultat de décision en utilisant leur théorème de Kuratowski généralisé. Comme l'ensemble des graphes de largeur arborescente au plus k est clos par minoration, il s'exprime par une famille d'obstructions \mathcal{F}_k . En construisant une décomposition approchée, ils arrivent à tester efficacement si le graphe a pour mineur un des éléments de la famille \mathcal{F}_k . La complexité de cet algorithme est $O(n^2)$ mais celui-ci est non constructif. Bodlaender [Bod96] améliore ce résultat et donne un algorithme linéaire pour ce problème de décision ainsi que pour le problème de construction d'une décomposition arborescente correspondante. Cependant, les complexités mentionnées cachent toutes une constante au moins exponentielle en k .

Parallèlement à cela, Courcelle (par exemple [Cou89]) puis d'autres comme Arnborg et col. [ALS91] ou Borie et col. [BPT91] montrent que certains problèmes difficiles peuvent se résoudre efficacement pour les graphes de largeur arborescente bornée : tout problème pouvant s'exprimer dans une logique monadique du second ordre étendue se résout en temps linéaire² ; ces algorithmes cachent cependant eux aussi une constante exponentielle en la largeur arborescente. Parmi ces problèmes, nous trouvons le problème du stable maximum et celui du cycle hamiltonien.

Ces résultats stimulent le travail sur le calcul de la largeur arborescente et sur les triangulations des graphes. En effet, comme nous l'avons déjà mentionné, trouver une bonne décomposition arborescente revient à trouver une bonne triangulation. Le calcul de la largeur arborescente est montré polynomial pour les graphes d'intervalles circulaires [SSP94], les graphes de permutation [BKK95] ou les graphes de cordes [Klo96]. Plus généralement, en étudiant les triangulations minimales des graphes, Bouchitté et Todinca [BT01a, BT01b] montrent que ce calcul se fait en temps polynomial pour toute classe de graphes ayant un nombre polynomial de séparateurs minimaux.

Dans la suite de leur série d'articles sur les mineurs de graphes, Robertson et Seymour [RS91] présentent un nouveau type de décompositions de graphes proche des décompositions arborescentes : les décompositions en branches qui, comme les décompositions arborescentes, sont associées à un paramètre :

²Par exemple, le problème de 3-coloration d'un graphe s'exprime par la formule :

$$\begin{aligned} &\exists A, B, C \subseteq V (\forall v \in V (v \in A) \vee (v \in B) \vee (v \in C)) \wedge \\ &(\forall uv \in E \neg(u \in A \wedge v \in A) \vee \neg(u \in B \wedge v \in B) \vee \neg(u \in C \wedge v \in C)). \end{aligned}$$

la largeur de branches notée $\text{bw}(G)$. Ils les introduisent comme obstruction aux décompositions arborescentes et montrent que pour les graphes,

$$\text{bw}(G) \leq \text{tw}(G) + 1 \leq \lfloor 3 \text{bw}(G)/2 \rfloor.$$

Les décompositions arborescentes et les décompositions en branches présentent de fortes similitudes. L'inégalité de Robertson et Seymour permet d'étendre les résultats du type de ceux de Courcelle aux graphes de largeur de branches au plus k ; des propriétés analogues à celles de la largeur arborescente sont ainsi établies pour la largeur de branches. Bodlaender et Thilikos [BT97] obtiennent un résultat analogue à celui de Bodlaender [Bod96] en donnant un algorithme linéaire pour décider si la largeur de branches est égale à k et construire le cas échéant une décomposition en branches correspondante. Kloks et col. [KKM99] montrent que le problème de décision associé à la largeur de branches est NP-complet.

Tous ces résultats corroborent l'idée que les décompositions arborescentes et les décompositions en branches possèdent les mêmes propriétés. Cependant, Kloks et col. [KKM99] montrent que le problème de décision associé à la largeur de branches est NP-complet pour les graphes triangulés alors que le problème correspondant pour la largeur arborescente est linéaire. D'autre part, Seymour et Thomas [ST94] donnent un algorithme polynomial de calcul de la largeur de branches pour les graphes planaires. Or le problème correspondant pour la largeur arborescente est toujours ouvert.

Au cours de cette thèse, nous explorons plus avant les liens mais aussi les différences entre les décompositions arborescentes et les triangulations des graphes d'une part et les décompositions en branches d'autre part. Les liens que nous mettons en avant permettent de transférer certains théorèmes d'un domaine à l'autre.

1.1 Résumé de la thèse

Cette thèse est organisée de la façon suivante.

Dans le chapitre 2, nous précisons la terminologie et les notations usuelles sur les graphes et les hyper-graphes que nous utilisons par la suite. Nous introduisons ensuite des notions spécifiques à notre champ d'étude. Nous commençons par celle de *séparateur minimal* avant de nous intéresser à celle, plus générale, de *séparation* et d'étendre à ces objets la relation usuelle de parallélisme sur les séparateurs minimaux. Nous présentons ensuite une classe d'hyper-graphes étroitement liés aux séparateurs minimaux : les *hyper-graphes triangulés*. Approfondir ces liens nous amène à présenter certains théorèmes qui soulignent la structure arborescente de ces hyper-graphes.

Le chapitre 3 poursuit l'étude des séparateurs minimaux en vue de leur énumération. Nous donnons un résultat de structure sur l'ensemble des a, b -séparateurs minimaux, puis nous présentons le meilleur algorithme d'énumé-

ration des séparateurs minimaux à ce jour [BBC00]. Ensuite, nous proposons un nouvel algorithme qui, s'il ne représente pas une avancée dans le cas général, peut s'adapter plus facilement à certaines classes d'hyper-graphes, ce que nous faisons au chapitre 8 pour obtenir un algorithme spécifique aux hyper-graphes planaires plus efficace que les algorithmes déjà connus.

Au chapitre 4, nous choisissons de présenter les *décompositions arborescentes* et les *décompositions en branches* comme provenant de certains schémas « diviser pour régner » sur les hyper-graphes suggérant ainsi que ces deux types de décompositions ne sont que deux facettes d'un même objet plus général. Pour confirmer cette intuition, nous introduisons deux outils permettant de manipuler les décompositions en branches : les *matriochkas* et les *arbres de matriochkas*. À l'aide de ces outils, nous pouvons associer des décompositions en branches à des décompositions arborescentes et réciproquement, et ainsi donner une preuve d'un théorème min/max de Robertson et Seymour reliant la largeur de branches et la largeur arborescente d'un hyper-graphe. Nous terminons ce chapitre en esquissant un lien entre ces deux types de décompositions et les *triangulations des hyper-graphes*.

En étudiant certaines triangulations, Bouchitté et Todinca obtiennent un algorithme général de calcul de la largeur arborescente. Les chapitres 5, 6 et 7 montrent comment leur démarche se généralise aux décompositions en branches. Plus précisément, le chapitre 5 présente un algorithme de calcul de triangulations dites *serrées* lequel est utilisé au chapitre 6 pour retrouver le résultat originel de Bouchitté et Todinca d'une part et pour obtenir un algorithme de calcul de largeurs de branches d'autre part. Pour montrer que les conditions d'utilisations de l'algorithme de triangulation s'appliquent, nous nous restreignons à une sous-classe de décompositions en branches. Pour la définir nous utilisons la notion de *profil*. Une fois l'algorithme de Bouchitté et Todinca généralisé, nous l'appliquons au chapitre 7 à certaines classes de graphes. Nous montrons que le calcul de la largeur de branches est polynomial pour les graphes d'intervalles circulaires, les graphes de nombre astéroïde borné (sous certaines conditions) et d'autres classes. De cette façon, nous retrouvons certains résultats antérieurs que nous étendons.

Le chapitre 8 est dédié à l'étude des graphes et des hyper-graphes planaires. Après quelques considérations topologiques, nous introduisons le *graphe intermédiaire* associé à un hyper-graphe plan dont certains cycles permettent de représenter certaines frontières et certains séparateurs minimaux. Cette caractérisation des séparateurs minimaux nous permet de spécialiser l'algorithme introduit au chapitre 3 pour obtenir un algorithme efficace d'énumération des séparateurs minimaux des graphes planaires. Ce graphe permet aussi de représenter certaines décompositions en branches de façon simple. De plus, comme le graphe intermédiaire d'un hyper-graphe plan et celui de son dual sont les mêmes, nous pouvons déduire d'une décomposition en branches d'un hyper-graphe planaire une décomposition de son dual. De cette façon, nous montrons que la largeur arborescente d'un

graphe et celle de son dual diffèrent d'au plus un.

Le chapitre 9 nous permet de revenir sur les résultats de cette thèse tout en proposant des extensions possibles.

Sommaire

2.1 Terminologie	7
2.2 Séparateurs minimaux et frontières	10
2.2.1 Séparateurs minimaux	10
2.2.2 Frontières	13
2.3 Graphes et hyper-graphes triangulés	15
2.3.1 Séparateurs minimaux des hyper-graphes triangulés	16
2.3.2 Théorèmes de caractérisation	17
2.3.3 Cliques maximales des hyper-graphes triangulés	20

Cette thèse se place dans le cadre des hyper-graphes. Nous aurions pu nous restreindre à celui des graphes. En effet, les hyper-graphes peuvent paraître plus difficiles à manipuler. Cependant non seulement leur introduction ne complique pas les démonstrations – ce point seul pourrait justifier leur introduction – mais elle les simplifie notablement.

Notre étude traite de « décompositions ». Nous allons donc naturellement être amenés à « décomposer ». Nous intéresser à des hyper-graphes permet des preuves par induction du type (voir fig. 2.1) :

- « Soit une partition (E_1, E_2) de l'ensemble E des hyper-arêtes. Considérons l'hyper-graphe G_1 obtenu à partir de G en conservant les hyper-arêtes de E_1 et en contractant celles de E_2 en une nouvelle hyper-arête. De même considérons G_2 obtenu en conservant les hyper-arêtes de E_2 . Par induction, la propriété est vraie pour G_1 et G_2 ...
- ... donc la propriété est vraie. »

Ce type de démonstrations ne s'adapte aux graphes qu'à l'aide d'artifices *ad-hoc* destinés à contourner l'utilisation d'hyper-arêtes : les hyper-graphes constituent le cadre bien adapté à notre étude.

Nous considérons des hyper-graphes finis, non orientés et connexes.

2.1 Terminologie

Avant toute chose, il est bon de poser clairement les définitions que nous utilisons par la suite. Les notions que nous introduisons ici sont pour la plupart très classiques pour les graphes mais certaines présentent des subtilités quand nous les adaptions aux hyper-graphes. De plus, la notion même d'hyper-graphe que nous utilisons n'est pas la notion usuelle.

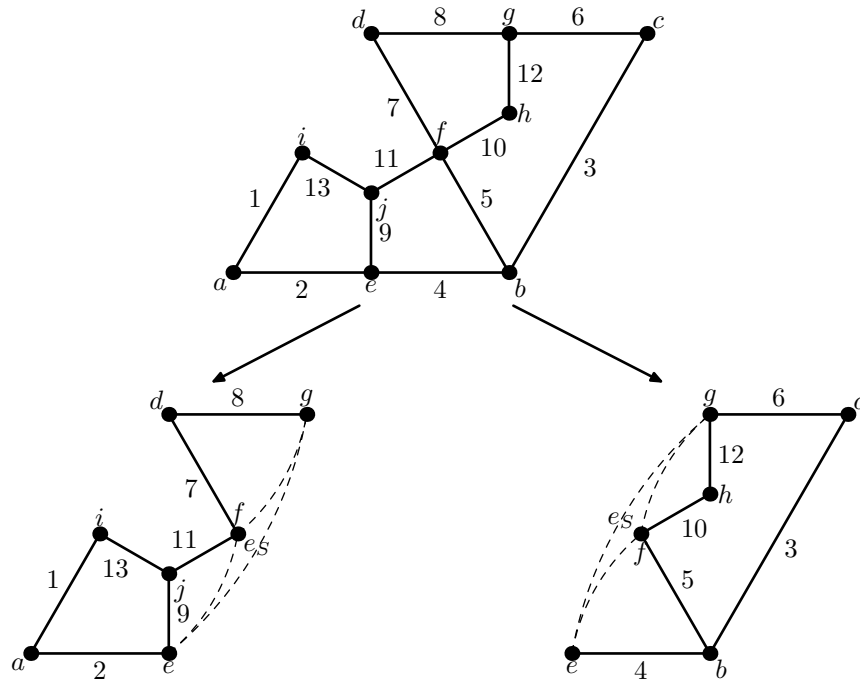


FIG. 2.1 – Utilisation des hyper-graphes pour les démonstrations

Définitions 2.1 (Hyper-graphe, graphe, graphe d'incidence)

Un hyper-graphe est un couple que nous notons généralement $G = (V, E)$. Les ensembles V et E sont finis. Les éléments de V sont les sommets de G et ceux de E sont les hyper-arêtes de G . Chaque élément de E est un multi-ensemble de sommets de G ; c'est-à-dire un ensemble de sommets qui peut contenir plusieurs fois le même sommet.

La taille d'une plus grosse hyper-arête de G est notée $\gamma(G)$ si les sommets sont comptés avec multiplicité et $\beta(G)$ sinon. Si G est un hyper-graphe, nous notons $V(G)$ et $E(G)$ ses ensembles de sommets et d'hyper-arêtes.

Un graphe est un hyper-graphe dont toutes les hyper-arêtes sont de taille deux. Nous notons xy l'arête $\{x, y\}$.

Soit $G = (V, E)$ un hyper-graphe. Le multigraphe dont les sommets sont les éléments de $V \cup E$ et dont les arêtes sont les couples (x, e) où e est une hyper-arête de G et où x est une extrémité de e est le graphe d'incidence de G .

La définition d'hyper-graphe que nous donnons correspond plutôt à des « multi-hyper-graphes ». Mais le plus souvent, nous pouvons ne considérer que des « vrais » hyper-graphes, c'est-à-dire des hyper-graphes dans lesquels les hyper-arêtes sont simplement des ensembles de sommets. En effet, si nous retirons les occurrences multiples de sommets dans les « multi-hyper-arêtes » d'un « multi-hyper-graphe », nous obtenons un « vrai » hyper-

graphe. Le plus souvent, nous pouvons identifier un « multi-hyper-graphe » à son hyper-graphe induit. Cependant, au chapitre 8, cette identification ne fonctionne plus car la notion de dualité dans les (hyper-)graphes planaires fait naturellement apparaître des multi-(hyper-)arêtes que nous ne pouvons pas ignorer.

Définitions 2.2 (Adjacence, incidence, voisinage, degré)

Deux sommets distincts d'un hyper-graphe G sont adjacents s'ils appartiennent à une même hyper-arête. Une hyper-arête est incidente à ses sommets ; elle relie ses sommets. Le voisinage d'un sommet x de V est l'ensemble $N_G(x)$ (ou simplement $N(x)$) des sommets qui lui sont adjacents. Plus généralement, pour X inclus dans $V(G)$, nous notons $N_G(X)$ le voisinage de X c'est-à-dire l'ensemble des sommets de $V \setminus X$ adjacents à au moins un sommet de X . Le degré d'un sommet est le cardinal de son voisinage.

Définitions 2.3 (Sous-structures et sur-structures)

Soit $G = (V, E)$ un hyper-graphe.

L'hyper-graphe G' égal à (V, E') avec E' inclus dans E est un hyper-graphe partiel de G . Dans ce cas G est un sur-hyper-graphe de G' .

Si V' est inclus dans V , le sous-hyper-graphe de G induit par l'ensemble de sommets V' est l'hyper-graphe $G[V']$ égal à (V', E') avec E' , l'ensemble des hyper-arêtes de E dont toutes extrémités sont dans V' . De façon analogue, si E' est inclus dans E , le sous-hyper-graphe de G induit par E' est l'hyper-graphe $G[E']$ égal à (V', E') avec V' l'ensemble des extrémités des hyper-arêtes de E' . Un hyper-graphe de cette forme est un sous-hyper-graphe de G induit par l'ensemble V' ou l'ensemble E' . Un hyper-graphe G' est un sous-hyper-graphe partiel de G si c'est un hyper-graphe partiel d'un sous-hyper-graphe de G .

Les notions de sous-graphe, sous-graphe partiel et de sur-graphe sont les notions analogues pour les graphes.

Définitions 2.4 (Chaîne, cycle, corde, longueur)

Une chaîne μ d'un hyper-graphe G est une suite (x_1, \dots, x_l) de sommets de G telle que deux sommets consécutifs de μ soient toujours adjacents. Une chaîne est élémentaire si elle ne passe pas deux fois par le même sommet. Si les sommets x_l et x_1 sont égaux, μ est un cycle. Un cycle (x_1, \dots, x_l) est élémentaire si la chaîne (x_1, \dots, x_{l-1}) est élémentaire. Une corde d'une chaîne ou d'un cycle μ est une hyper-arête qui relie deux sommets non consécutifs sur μ . La longueur d'une chaîne est le nombre de sommets que contient cette chaîne moins un.

Définitions 2.5 (Connexité, composantes connexes)

Soit G un hyper-graphe. La relation « il existe une chaîne passant par les sommets x et y » est une relation d'équivalence. Les classes d'équivalence pour cette relation sont les composantes connexes de G . Un hyper-graphe est connexe s'il ne possède qu'une seule composante connexe.

Définitions 2.6 (Arbre, arbre enraciné, nœud, père, fils)

Un arbre est un graphe connexe sans cycle. Nous appelons nœuds les sommets d'un arbre. Un arbre enraciné est un arbre dont un sommet – la racine – est particularisé. Un tel arbre induit une relation d'ordre partiel sur ses nœuds. Le plus petit antécédent et les plus grands successeurs d'un nœud x , s'ils existent, sont respectivement le père et les fils de x .

Nous notons généralement les arbres T et les arbres enracinés $*T$.

Définitions 2.7 (Clique, cliques maximales)

Une clique est un hyper-graphe dans lequel tous les sommets sont deux à deux adjacents. Une clique d'un hyper-graphe est un sous-graphe qui est une clique. Une clique maximale d'un hyper-graphe G est une clique de G maximale pour l'inclusion.

2.2 Séparateurs minimaux et frontières

Comme nous l'avons déjà mentionné, l'objet de cette thèse est l'étude de certaines décompositions des hyper-graphes. Pour décomposer les hyper-graphes, nous allons utiliser les notions de séparateur minimal et de séparation que nous introduisons ici.

2.2.1 Séparateurs minimaux**Définitions 2.8 (Séparateur minimal)**

Soit G un hyper-graphe. Pour a et b deux sommets de G , un ensemble S est un a, b -séparateur de G si a et b ne sont pas dans une même composante connexe de $G \setminus S$. Un a, b -séparateur minimal de G est un a, b -séparateur de G qui est minimal pour l'inclusion. Un ensemble S est un séparateur minimal de G s'il existe deux sommets a et b de G tels que S soit un a, b -séparateur minimal.

Nous notons Δ_G l'ensemble des séparateurs minimaux de G . Si Ω est inclus dans $V(G)$, $\Delta_G(\Omega)$ désigne l'ensemble des séparateurs minimaux de G inclus dans Ω .

Remarquons tout de suite que deux séparateurs minimaux peuvent être inclus l'un dans l'autre. Dans la figure 2.2 l'ensemble $\{e, f, g\}$ est un a, b -séparateur minimal, $\{f, g\}$ est un j, k -séparateur minimal et $\{g\}$ est un h, i -séparateur minimal.

Définition 2.9 (Composante pleine)

Soit S un ensemble de sommets d'un hyper-graphe G . Une composante connexe C de $G \setminus S$ est pleine par rapport à S si son voisinage $N(C)$ est égal à S . Nous notons $\mathcal{C}_G(S)$ l'ensemble des composantes connexes de $G \setminus S$. La composante connexe de $G \setminus S$ contenant un sommet a n'appartenant pas

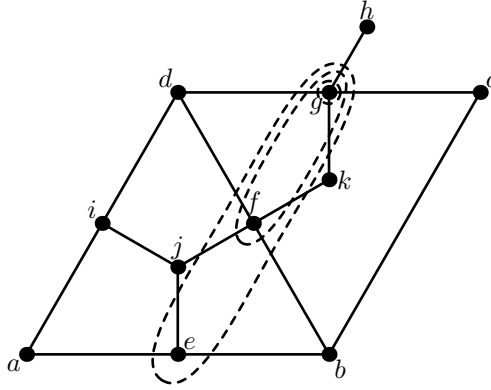


FIG. 2.2 – Séparateurs minimaux

à S est notée $C_G^a(S)$. Nous notons $\mathcal{C}_G^*(S)$ l'ensemble des composantes pleines par rapport à S .

Dans la figure 2.2, les composantes connexes de $G \setminus S$ sont $\{a, d, i, j\}$, $\{b, c\}$, $\{h\}$ et $\{k\}$. Les composantes pleines sont $\{a, d, i, j\}$ et $\{b, c\}$.

Nous utilisons beaucoup la caractérisation suivante des séparateurs minimaux à l'aide de composantes pleines.

Propriété 2.10 Soient G un hyper-graphe, S un ensemble de sommets de G et C et D deux composantes connexes de $\mathcal{C}_G(S)$. Les propositions suivantes sont équivalentes :

- (i) C et D sont pleines par rapport à S ;
- (ii) il existe a dans C et b dans D tels que S soit un a, b -séparateur minimal ;
- (iii) pour tout a dans C et b dans D , S est un a, b -séparateur minimal.

□ Montrons que $i \Rightarrow iii$. Soient a dans C , b dans D et v dans S . Comme C est une composante connexe pleine de $G \setminus S$ par rapport à S , il existe une chaîne $p_{a \rightarrow v}$ de a à v dans $C \cup \{v\}$. De même, il existe une chaîne $p_{v \rightarrow b}$ dans $D \cup \{v\}$. En concaténant $p_{a \rightarrow v}$ et $p_{v \rightarrow b}$, nous obtenons une chaîne de a à b dans $C \cup D \cup \{v\}$. Par conséquent $S \setminus \{v\}$ n'est pas un a, b -séparateur. Le sommet v étant quelconque, S est bien un séparateur minimal.

L'implication $iii \Rightarrow ii$ est évidente.

Montrons l'implication $ii \Rightarrow i$. Soit a dans C et b dans D tels que S soit un a, b -séparateur minimal. Le voisinage de C sépare tout sommet de C de tout sommet de D . Par conséquent, $N(C)$ est un a, b -séparateur. Le a, b -séparateur S étant minimal, les ensembles S et $N(C)$ sont égaux. La composante C est donc pleine par rapport à S . De même, D est pleine par rapport à S . ■

Nous en déduisons directement :

Corollaire 2.11 *Soit G un hyper-graphe. Un ensemble de sommets S est un séparateur minimal de G si et seulement si il admet au moins deux composantes pleines.*

Dans le découpage récursif des hyper-graphes, les séparateurs que nous utilisons possèdent certaines propriétés de *parallélisme* que nous définissons ici.

Définitions 2.12 (Parallélisme de séparateurs minimaux)

Soient S et T deux ensembles de sommets d'un hyper-graphe G . Nous dirons que S est parallèle à T si S est inclus dans un ensemble de la forme $T \cup C$ où C est une composante connexe de $G \setminus T$. Dans le cas contraire, c'est-à-dire si S rencontre au moins deux composantes connexes de $G \setminus T$, nous dirons que S croise T .

Lemme 2.13 *Les relations de croisement et de parallélisme de séparateurs minimaux sont symétriques.*

□ Notons d'abord que si la relation de croisement est symétrique, alors celle de parallélisme l'est aussi.

Soient S et T deux séparateurs minimaux d'un hyper-graphe G et supposons que S croise T . Montrons alors que T croise S . Soient a et b deux sommets de S qui se trouvent dans deux composantes connexes de $G \setminus T$. D'après la propriété 2.10, il existe C et D deux composantes pleines dans $\mathcal{C}_G^*(S)$. Par définition, a et b sont dans le voisinage de C . Il existe donc une chaîne $\mu_{a \rightarrow b}^C$ de a à b dans $C \cup \{a, b\}$.

Comme T est un a, b -séparateur, il intersecte la chaîne $\mu_{a \rightarrow b}^C$ et donc la composante connexe C . Les composantes connexes C et D jouant un rôle symétrique, T intersecte aussi D et donc T croise S . ■

La propriété suivante fournit une caractérisation du parallélisme pour les séparateurs minimaux.

Propriété 2.14 *Soient S et T deux séparateurs minimaux d'un hyper-graphe G . Deux séparateurs minimaux S et T sont parallèles si et seulement si il existe une composante connexe C_T de $G \setminus T$ telle que S soit inclus dans $C_T \cup N(C_T)$.*

□ Soient S et T deux séparateurs minimaux d'un hyper-graphe G . Si S est inclus dans $C_T \cup N(C_T)$ où C est une composante connexe de $G \setminus T$, alors S croise au plus une composante connexe de $G \setminus T$: S est parallèle à T .

Réciproquement, supposons que S soit parallèle à T . Soient C_S et D_S deux composantes pleines dans $\mathcal{C}_G^*(S)$. Comme, d'après le lemme 2.13, la relation de parallélisme est symétrique, T est parallèle à S et ne peut pas rencontrer à la fois C_S et D_S , nous pouvons donc supposer que C_S et T sont disjoints. Puisque l'ensemble C_S induit un sous-hyper-graphe connexe de G

et que C_S et T sont disjoints, C_S est inclus dans une composante connexe C_T de $G \setminus T$. Comme S est le voisinage de C_S , S est inclus dans $C_T \cup N(C_T)$. ■

2.2.2 Frontières

Les séparateurs minimaux constituent un outil puissant pour l'étude des décompositions arborescentes. Cependant, ils ne sont pas suffisants pour celle des décompositions en branches. C'est pourquoi nous préférons souvent à cette notion celle, plus générale, de *frontière*.

Définitions 2.15 (Frontière, séparation)

Soit G un hyper-graphe. La frontière d'un ensemble d'hyper-arêtes X est l'ensemble $\partial_G(X)$ (ou plus simplement $\partial(X)$) des sommets incidents à une hyper-arête de X et à une de $E(G) \setminus X$.

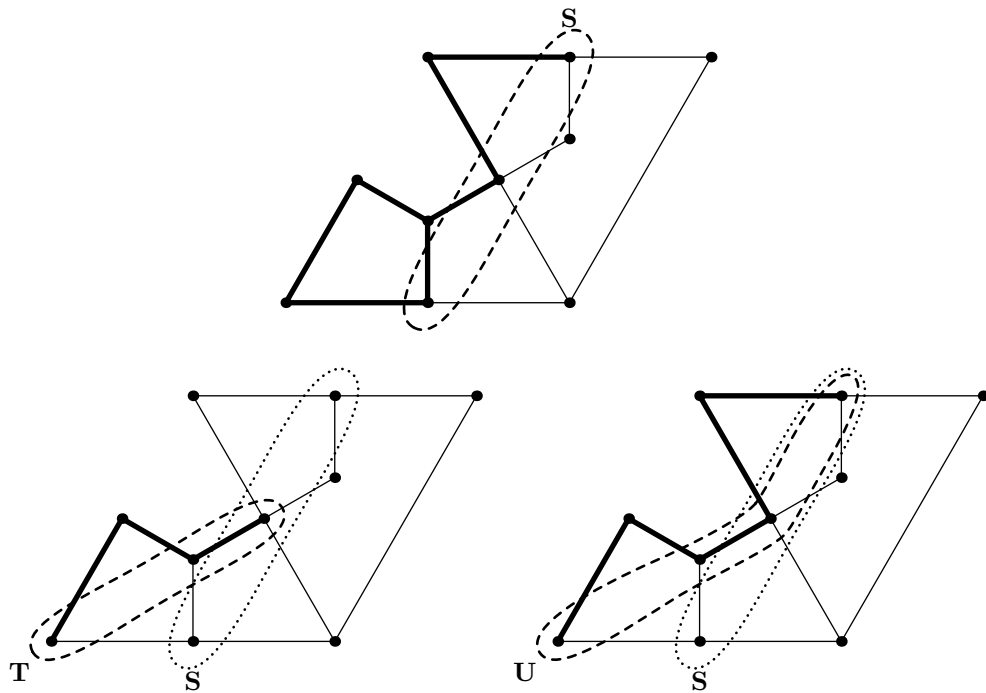
Une séparation d'un hyper-graphe G est une partition de $E(G)$ en deux éléments $\{E_1, E_2\}$. La frontière d'une séparation est la frontière d'un de ses éléments. Une frontière est un ensemble de sommets qui est la frontière d'une séparation.

Propriété 2.16 Soit S un séparateur minimal d'un hyper-graphe G . Il existe une séparation de G dont la frontière est S .

□ Soit C une composante pleine de $\mathcal{C}_G^*(S)$. Notons A l'ensemble des hyper-arêtes de G incidentes à au moins un sommet de C . La composante C étant pleine, chaque sommet de S est incident à au moins une hyper-arête de A . De plus, comme S est un séparateur minimal, il admet au moins une seconde composante pleine D . L'ensemble $E \setminus A$ contient toutes les hyper-arêtes incidentes à D donc tout sommet de S est incident à une hyper-arête de $E \setminus A$; la frontière de A est S . ■

La notion de frontières étend donc bien celle de séparateurs minimaux mais pour les utiliser, nous devons encore les munir d'une notion de parallélisme qui étende celle définie pour les séparateurs minimaux. Pour cela, nous pourrions utiliser la définition 2.12 ou la propriété 2.14. Ces notions sont bien distinctes comme l'illustre la figure 2.3. Cependant l'extension basée sur la définition initiale est trop faible pour l'utilisation que nous souhaitons en faire. À l'inverse, la propriété 2.14 conduit à une notion trop forte comme le justifie la propriété 6.24.

Nous introduisons donc une notion de parallélisme intermédiaire qui étend la notion de parallélisme pour les séparateurs minimaux tout en palliant aux inconvénients mentionnés ci-dessus.



La frontière T est parallèle à la frontière S au sens de la définition 2.12 et de la propriété 2.14.

La frontière U est parallèle à la frontière S au sens de la définition 2.12 mais pas au sens de la propriété 2.14.

FIG. 2.3 – Deux notions possibles de parallélisme de frontières

Définitions 2.17 (Parallélisme de frontières)

Soient S et T deux ensembles de sommets d'un hyper-graphe G . Nous dirons que S est parallèle à T si S est inclus dans T ou si S est inclus dans $C \cup N(C)$ où C est une composante connexe de $G \setminus T$.

Remarque 2.18 La notion de parallélisme que nous venons de définir n'est pas symétrique pour les frontières.

Les frontières étant issues d'ensembles d'hyper-arêtes, nous définissons aussi une relation de chevauchement pour les ensembles d'hyper-arêtes.

Définition 2.19 (Chevauchement)

Deux sous-ensembles A et B de X se chevauchent (dans X) si les ensembles $A \cap B$, $A \setminus B$, $B \setminus A$ et $X \setminus (A \cup B)$ sont tous non vides. S'ils ne se chevauchent pas, ils vérifient l'une des quatre conditions suivantes : $A \subseteq B$, $B \subseteq A$, $A \cap B = \emptyset$ ou $A \cup B = X$. Quand il n'y a pas d'ambiguïté, nous disons seulement les ensembles « A et B se chevauchent » ou « A et B ne se chevauchent pas ».

Lemme 2.20 Si les ensembles A et B ne se chevauchent pas dans X , les ensembles A et $X \setminus B$ ne se chevauchent pas non plus.

□ Soient A , B inclus dans X ne se chevauchant pas.

- si $A \subseteq B$, alors $A \cap (X \setminus B) = \emptyset$;
- si $B \subseteq A$, alors $A \cup (X \setminus B) = X$;
- si $A \cap B = \emptyset$, alors $A \subseteq (X \setminus B)$;
- si $A \cup B = X$, alors $(X \setminus B) \subseteq A$.

Dans tous les cas, les ensembles A et $X \setminus B$ ne se chevauchent pas. ■

Puisque la relation de chevauchement est stable par passage au complémentaire, nous pouvons l'étendre aux séparations en posant que $\mathcal{E} = (E_1, E_2)$ chevauche $\mathcal{F} = (F_1, F_2)$ si l'un des éléments de \mathcal{E} chevauche l'un des éléments de la séparation \mathcal{F} .

2.3 Graphes et hyper-graphes triangulés

Les décompositions d'hyper-graphes que nous considérons peuvent être vues comme des plongements de l'hyper-graphe en entrée dans un hyper-graphe triangulé. Nous présentons ici les propriétés de base des graphes et hyper-graphes triangulés.

Définitions 2.21 (Hyper-graphe triangulé, triangulation)

Un hyper-graphe est triangulé si tout cycle ayant au moins quatre sommets admet une corde. Une triangulation H d'un hyper-graphe G est un hyper-graphe triangulé ayant le même ensemble de sommets et tel que toute hyper-arête de G soit incluse dans une hyper-arête de H .

2.3.1 Séparateurs minimaux des hyper-graphes triangulés

Les séparateurs minimaux des hyper-graphes triangulés ont une forme très particulière; celle-ci fournit une première caractérisation des hyper-graphes triangulés due à Dirac.

Théorème 2.22 ([Dir61])

Un hyper-graphe est triangulé si et seulement si tous ses séparateurs minimaux induisent des cliques.

□ Soient G un hyper-graphe triangulé et S un séparateur minimal de G . Montrons que S induit une clique de G . Pour cela, soient x et y deux sommets quelconques de S . La propriété 2.11 assure qu'il existe deux composantes connexes pleines C et D de $G \setminus S$ par rapport à S . Les sommets x et y possèdent donc des voisins v_x et v_y dans C . De plus, comme les sommets v_x et v_y appartiennent à une même composante connexe, il existe une chaîne (x, c_1, \dots, c_p, y) de x à y dont tous les sommets c_i appartiennent à C . Si nous choisissons une telle chaîne de longueur minimale, la seule corde qu'elle puisse admettre relie x et y .

Choisissons de manière analogue une chaîne (y, d_1, \dots, d_q, x) dont tous les sommets d_i soient dans D . Le cycle $\mu = (x, c_1, \dots, c_p, y, d_1, \dots, d_q, x)$ est de longueur au moins quatre. L'hyper-graphe G étant triangulé, μ possède une corde. Les sommets c_i et d_j n'appartenant pas à la même composante connexe de $G \setminus S$, les sommets c_i ne sont pas adjacents aux sommets d_j . La seule corde que puisse admettre le cycle μ relie les sommets x et y . Les sommets x et y étant quelconques, S induit une clique de G .

Réciproquement, si G n'est pas triangulé, il possède un cycle μ égal à (x_1, \dots, x_p, x_1) de longueur au moins quatre sans corde. Nous allons construire un x_1, x_3 -séparateur minimal qui n'induit pas de clique de G . L'ensemble $N(x_1)$ est un x_1, x_3 -séparateur. Nous pouvons donc en extraire un x_1, x_3 -séparateur minimal S . Ce séparateur minimal contient x_2 . De plus, comme aucun des sommets de x_4 à x_{p-1} n'est voisin de x_1 , S contient aussi x_p . Les sommets x_2 et x_p ne sont pas voisins donc S n'induit pas de clique. ■

La forme particulière des séparateurs minimaux des hyper-graphes triangulés impose une structure forte sur leur ensemble.

Propriété 2.23 *Les séparateurs minimaux d'un hyper-graphe triangulé sont deux à deux parallèles.*

□ Soient G un hyper-graphe triangulé et S et T deux séparateurs minimaux de G . Comme un séparateur minimal ne peut séparer deux sommets d'une clique et que, d'après le théorème 2.22, S et T induisent des cliques de G , T ne rencontre qu'une composante connexe de $G \setminus S$ ce qui prouve que les séparateurs minimaux S et T sont deux à deux parallèles. ■

2.3.2 Théorèmes de caractérisation

Les deux caractérisations suivantes des graphes triangulés apparaissant dans les travaux de Gavril [Gav74] ; elles soulignent la structure arborescente des graphes et des hyper-graphes triangulés (voir fig. 2.4).

Définition 2.24 (Arbre des cliques)

Un arbre des cliques d'un hyper-graphe G est un arbre \mathcal{T} dont les nœuds sont étiquetés par les cliques maximales de G et tel que si u et v sont deux nœuds de \mathcal{T} dont les cliques correspondantes Ω_u et Ω_v ont une intersection non vide, alors $\Omega_u \cap \Omega_v$ est contenue dans toutes les cliques étiquetant les nœuds sur la chaîne de T entre u et v .

Nous pouvons reformuler cette définition d'arbre des cliques en disant que l'ensemble des étiquettes de T contenant un sommet x de G induit un sous-arbre de T .

Théorème 2.25 ([Gav74])

Un hyper-graphe est triangulé si et seulement si il admet un arbre des cliques.

La seconde caractérisation fait intervenir la notion de graphe d'intersections.

Définition 2.26 (Graphe d'intersections)

Soient Π un ensemble donné et \mathcal{F} une famille finie de sous-ensembles de Π . Le graphe d'intersection associé à \mathcal{F} est le graphe $G_{\mathcal{F}}$ obtenu en associant un sommet à chaque élément de \mathcal{F} et en plaçant une arête entre chaque paire de sommets correspondant à deux éléments de \mathcal{F} dont l'intersection est non vide. La famille \mathcal{F} est un modèle d'intersections du graphe $G_{\mathcal{F}}$.

Tout graphe G est un graphe d'intersections ; il suffit de choisir l'ensemble Π égal à $E(G)$ et d'associer à chaque sommet de G l'ensemble des arêtes auxquelles il est incident. Cependant, certaines classes de graphes sont issues de modèles « plus naturels » comme des modèles d'intersections de figures géométriques et possèdent souvent des propriétés intéressantes issues de ces modèles. C'est le cas des graphes triangulés dont la seconde caractérisation énonce qu'ils sont les graphes d'intersections d'une famille de sous-arbres d'un arbre.

Théorème 2.27 ([Gav74])

Un graphe est triangulé si et seulement si c'est le graphe d'intersections d'une famille de sous-arbres d'un arbre.

La caractérisation 2.22 de Dirac des hyper-graphes triangulés est une caractérisation locale : un séparateur minimal doit induire une clique. En revanche, la caractérisation des hyper-graphes triangulés à l'aide d'un arbre des cliques 2.25 est une caractérisation globale. Ho et Lee [HL89] puis, de manière indépendante, Lundquist [Lun90] relient ces deux caractérisations

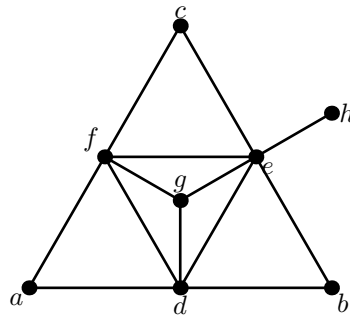
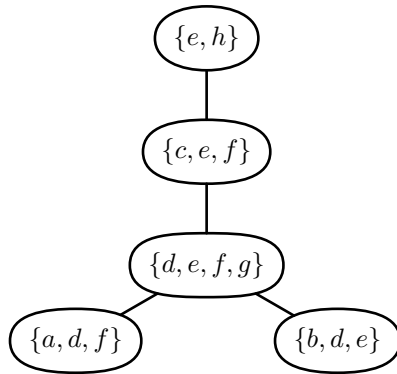
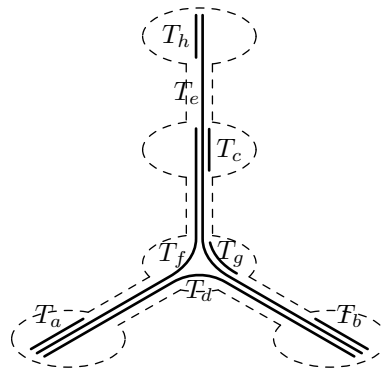
Un graphe triangulé G Un arbre des cliques de G Le graphe G comme graphe d'intersections des sous-arbres d'un arbre

FIG. 2.4 – Un graphe triangulé et deux de ses représentations

en montrant comment obtenir les séparateurs minimaux d'un hyper-graphe triangulé H à l'aide d'un arbre des cliques de H .

Théorème 2.28 ([HL89, Lun90])

Soit H un hyper-graphe triangulé et \mathcal{T} un arbre des cliques quelconque de H . Un ensemble S est un séparateur minimal de H si et seulement si S est égal à $\Omega_1 \cap \Omega_2$ où Ω_1 et Ω_2 sont deux cliques maximales adjacentes dans \mathcal{T} .

Pour démontrer ce théorème, nous avons besoin du lemme suivant :

Lemme 2.29 Soient H un hyper-graphe triangulé, \mathcal{T} un arbre des cliques de H , $\Omega_1 \Omega_2$ une arête de \mathcal{T} , T_{Ω_1} et T_{Ω_2} les sous-arbres de $\mathcal{T} \setminus \{\Omega_1 \Omega_2\}$ contenant respectivement les nœuds Ω_1 et Ω_2 et V_{Ω_1} et V_{Ω_2} les ensembles des sommets de H appartenant respectivement aux étiquettes de T_{Ω_1} et de T_{Ω_2} .

L'ensemble S égal à $\Omega_1 \cap \Omega_2$ sépare tous les sommets de $V_{\Omega_1} \setminus S$ et $V_{\Omega_2} \setminus S$.

□ Soient a et b respectivement dans $V_{\Omega_1} \setminus S$ et $V_{\Omega_2} \setminus S$ et $\mu_{a \rightarrow b}$ une chaîne (x_1, \dots, x_p) de a à b dans H . La réunion des sous-arbres de \mathcal{T} correspondant aux sommets x_i induit un sous-arbre $T_{\mu_{a \rightarrow b}}$ de \mathcal{T} . Comme a appartient à $V_{\Omega_1} \setminus S$, le sous-arbre T_a est inclus dans T_{Ω_1} . De même, T_b est inclus dans T_{Ω_2} . Par conséquent, $T_{\mu_{a \rightarrow b}}$ contient l'arête $\Omega_1 \Omega_2$ ce qui implique qu'un des sous-arbres T_{x_i} passe par l'arête $\Omega_1 \Omega_2$. La chaîne $\mu_{a \rightarrow b}$ rencontre donc S ce qui prouve bien que S sépare a de b . ■

Nous pouvons revenir à la démonstration du théorème 2.28.

□ Montrons tout d'abord que si Ω_1 et Ω_2 sont deux cliques adjacentes dans \mathcal{T} , leur intersection S est un séparateur minimal. Soient a dans $\Omega_1 \setminus \Omega_2$ et b dans $\Omega_2 \setminus \Omega_1$. D'après le lemme 2.29, S est un a, b -séparateur. Comme S est inclus à la fois dans le voisinage de a et dans celui de b , les composantes $C_G^a(S)$ et $C_G^b(S)$ sont pleines ce qui prouve que S est un a, b -séparateur minimal.

Réciproquement, soient S un a, b -séparateur minimal et T_a et T_b les sous-arbres de \mathcal{T} induits respectivement par les étiquettes contenant a et b . Les sommets a et b n'étant pas adjacents, les arbres T_a et T_b sont disjoints. Soient Ω_a un nœud de T_a et Ω_b un nœud de T_b tels que la chaîne $(\Omega_1, \dots, \Omega_p)$ de Ω_a à Ω_b soit la plus courte possible. Notons S_i le séparateur $\Omega_i \cap \Omega_{i+1}$ et montrons que S appartient à l'ensemble \mathcal{S} des séparateurs S_i .

Il existe un séparateur S_i de \mathcal{S} qui est inclus dans S . Supposons que ce ne soit pas le cas, pour chaque i dans $[1, p-1]$, il existe un sommet x_i dans $S_i \setminus S$. Comme x_i et x_{i+1} appartiennent à Ω_{i+1} , ils sont incidents. Nous pouvons donc construire une chaîne $\mu_{a \rightarrow b}$ égale à $(a, x_1, \dots, x_{p-1}, b)$. Cette chaîne relie a et b et évite S , ce qui contredit le fait que S soit un a, b -séparateur. Il

existe un séparateur S_i inclus dans S . Mais d'après le lemme 2.29, S_i sépare a et b . Comme S est un a, b -séparateur minimal et qu'il contient S_i , il est égal à S_i ce qui achève la démonstration. ■

2.3.3 Cliques maximales des hyper-graphes triangulés

Le théorème de Lundquist, Ho et Lee permet de caractériser les séparateurs minimaux d'un hyper-graphe triangulé H à partir des cliques maximales de H . Il est aussi possible de donner une caractérisation des cliques maximales de H à l'aide des séparateurs minimaux de H . Ce sont ces liens que nous étudions dans cette partie.

Propriété 2.30 *Soit Ω une clique maximale d'un hyper-graphe triangulé H . Aucune composante connexe de $H \setminus \Omega$ n'est pleine par rapport à Ω .*

□ Nous allons démontrer cette propriété en raisonnant par l'absurde. Supposons qu'il existe une clique maximale Ω d'un hyper-graphe triangulé H qui admette une composante pleine C de $H \setminus \Omega$.

Notons $N_\Omega(x)$ le voisinage de x dans Ω , c'est-à-dire l'ensemble $N(x) \cap \Omega$ et montrons qu'il existe μ une chaîne (x_1, \dots, x_p) de C telle que les voisinages dans Ω des sommets de l'intérieur de la chaîne ne contiennent strictement ni $N_\Omega(x_1)$ ni $N_\Omega(x_p)$ et telle que $N_\Omega(x_1) \setminus N_\Omega(x_p)$ et $N_\Omega(x_p) \setminus N_\Omega(x_1)$ ne soient pas vides. À l'aide d'une telle chaîne de longueur minimale, nous allons construire un cycle sans corde de longueur au moins quatre.

Soit x dans C dont le voisinage dans Ω est maximal. Comme Ω est une clique maximale, $\Omega \cup \{x\}$ n'admet pas de clique. Il existe donc un sommet u dans $\Omega \setminus N_\Omega(x)$. Soient y un voisin de u dans C dont le voisinage dans Ω est maximal et $\mu_{x \rightarrow y}$ une chaîne entre x et y dans C . La maximalité de $N_\Omega(x)$ et $N_\Omega(y)$ assure qu'aucun voisinage $N_\Omega(x_i)$ ne les contient strictement. De plus, par construction, $N_\Omega(x) \setminus N_\Omega(y)$ et $N_\Omega(y) \setminus N_\Omega(x)$ sont non vides.

Considérons une telle chaîne μ la plus courte possible et un sommet x_i de l'intérieur de μ . Si $N_\Omega(x_i)$ est égal à $N_\Omega(x_1)$, alors en considérant la sous-chaîne (x_i, \dots, x_p) , nous obtenons une chaîne strictement plus courte que μ vérifiant les hypothèses requises. Les voisinages dans Ω des sommets de l'intérieur de μ ne contiennent pas $N_\Omega(x_1)$. De même, ils ne contiennent pas $N_\Omega(x_p)$. De plus, ils sont inclus dans $N_\Omega(x_1) \cap N_\Omega(x_p)$. En effet, si ce n'est pas le cas, quitte à « retourner » μ , nous pouvons supposer que $N_\Omega(x_i)$ n'est pas inclus dans $N_\Omega(x_1)$. Il existe donc un premier sommet x_{i_0} le long de μ dont le voisinage dans Ω n'est pas inclus dans $N_\Omega(x_1)$. La chaîne (x_1, \dots, x_{i_0}) est strictement plus courte que μ et vérifie les conditions requises contredisant ainsi le fait que μ soit la plus courte possible.

Soient u dans $N_\Omega(x_1) \setminus N_\Omega(x_p)$ et v dans $N_\Omega(x_p) \setminus N_\Omega(x_1)$. À l'aide de u , v et μ , nous pouvons définir le cycle λ égal à $(x_1, \dots, x_p, v, u, x_1)$. Par construction, il n'y a pas de corde entre deux sommets x_i et x_j de μ . De

plus, comme u n'appartient pas à $N_\Omega(x_p)$ et que $N_\Omega(x_p)$ contient chaque $N_\Omega(x_i)$ pour $i \geq 2$, u n'est adjacent à aucun sommet x_i pour $i \geq 2$. De même, v n'est adjacent à aucun sommet x_i pour $i < p$. Le cycle λ n'a donc pas de corde. Or sa longueur est au moins quatre ce qui est absurde car H est triangulé. ■

Lemme 2.31 *Soit H un hyper-graphe triangulé et S un séparateur minimal de H . Pour toute clique maximale Ω de H , il existe une composante C de $C_H(S)$ telle que $C \cup S$ contienne Ω .*

□ Comme un séparateur minimal ne peut pas séparer deux sommets adjacents, Ω n'intersecte qu'une composante connexe C dans $C_H(S)$. La clique est donc incluse dans $C \cup S$. ■

Remarque 2.32 Le lemme 2.31 montre que si Ω est une clique maximale d'un hyper-graphe triangulé H et si, pour chaque séparateur minimal S , nous choisissons une composante connexe C_S telle que Ω soit inclus dans $C_S \cup S$, alors la clique Ω est incluse dans l'intersection des ensembles $C_S \cup S$.

Ceci nous conduit à introduire les notions de famille de séparateurs voisins et de partie entre séparateurs voisins définies par Bouchitté et Todinca.

Définitions 2.33 (Famille de séparateurs voisins, bloc)

Soient $G = (V, E)$ un hyper-graphe et Ω inclus dans V .

Un séparateur minimal S inclus dans Ω borde Ω s'il existe une composante connexe C dans $C_G^*(S)$ telle que $C \cup S$ contienne Ω . Nous notons $\Gamma_G(\Omega)$ le sous-ensemble de $\Delta_G(\Omega)$ des séparateurs minimaux qui bordent Ω .

Une famille \mathcal{S} de séparateurs minimaux de G est une famille de séparateurs voisins³ si pour chaque séparateur S dans \mathcal{S} , il existe une composante connexe C_S dans $C_G^*(S)$ telle que $C_S \cup S$ contienne tous les séparateurs minimaux de \mathcal{S} . Si aucun séparateur minimal de \mathcal{S} ne contient les autres, la partie entre la famille \mathcal{S} de séparateurs voisins est l'ensemble $\mathcal{P}_G(\mathcal{S})$ égal à $\bigcap_{S \in \mathcal{S}} C_S \cup S$.

Un bloc de G est soit une partie entre séparateurs voisins, soit un ensemble de la forme $C \cup S$ avec C un élément de $C_G^*(S)$.

Lemme 2.34 *Soient H un hyper-graphe triangulé et Ω une clique maximale de H . Les ensembles $\Gamma_H(\Omega)$ et $\Delta_H(\Omega)$ sont égaux.*

□ Comme nous l'avons déjà dit, $\Gamma_H(\Omega)$ est un sous-ensemble de $\Delta_H(\Omega)$. Inversement, si S est un séparateur minimal de $\Delta_H(\Omega)$, le lemme 2.31 montre qu'il existe une composante connexe C de $C_H(S)$ telle que $C \cup S$ contienne

³Bien qu'elle soit mal choisie, nous gardons cette dénomination pour des raisons historiques.

Ω . De plus, comme Ω est une clique, et que S est strictement inclus dans Ω , il existe un sommet de $\Omega \setminus S$ dont le voisinage contient S . Par conséquent, la composante C est pleine ce qui prouve que S borde Ω . ■

La propriété suivante précise la remarque 2.32.

Théorème 2.35 ([Tod99])

Soient H un hyper-graphe triangulé et Ω une clique maximale de H . Si tous les éléments de $\Gamma_H(\Omega)$ sont inclus dans un séparateur minimal S de $\Gamma_H(\Omega)$, alors Ω est un bloc $C \cup S$ où C est une composante connexe de $C_H^(S)$. Dans le cas contraire, $\mathcal{P}_H(\Gamma_H(\Omega))$ existe et est égal à Ω .*

□ D'après le lemme 2.34, les ensembles $\Delta_H(\Omega)$ et $\Gamma_H(\Omega)$ sont égaux.

Plaçons-nous dans le premier cas, c'est-à-dire celui où un séparateur minimal S de $\Gamma_H(\Omega)$ contient tous les autres. D'après le lemme 2.31, il existe une composante connexe C de $C_H(S)$ telle que $C \cup S$ contienne Ω . De plus, comme S est inclus dans Ω , la composante C est pleine par rapport à S .

Supposons que Ω ne soit pas égal à $C \cup S$ et qu'il existe un sommet x dans $C \setminus \Omega$. Considérons dans un arbre des cliques de H une chaîne $\Omega, \Omega_1, \dots, \Omega_p$ de cliques adjacentes de sorte que x appartienne à Ω_p . Alors l'ensemble T égal à $\Omega \cap \Omega_1$ est un séparateur minimal inclus dans Ω . Le séparateur T est donc un élément de $\Gamma_H(\Omega)$ et par hypothèse, T est inclus dans S . Puisque x n'appartient pas à Ω , le lemme 2.29 permet d'affirmer que T sépare x de $\Omega \setminus T$. Comme T est inclus dans S , S sépare x de $\Omega \setminus S$ qui n'est pas vide (prop 2.30) contredisant le fait que Ω et x soient tous les deux inclus dans $C \cup S$. L'ensemble Ω est donc bien égal à $C \cup S$.

Plaçons-nous dans le second cas et supposons qu'aucun élément de $\Gamma_H(\Omega)$ ne contient tous les autres. Pour chaque séparateur minimal S de $\Gamma_H(\Omega)$, il existe d'après le lemme 2.31 une composante connexe C_S de $C_H(S)$ telle que $C_S \cup S$ contienne Ω . L'ensemble Ω est donc inclus dans $\mathcal{P}(\Gamma_H(\Omega))$.

Supposons que cette inclusion soit stricte et qu'il existe un sommet x dans $\mathcal{P}(\Gamma_H(\Omega)) \setminus \Omega$. Comme dans le premier cas, considérons une chaîne $\Omega, \Omega_1, \dots, \Omega_p$ de cliques maximales adjacentes d'un arbre des cliques de H telle que x appartienne à Ω_p . Le séparateur T égal à $\Omega \cap \Omega_1$ est un élément de $\Gamma_H(\Omega)$. D'après la propriété 2.30, $\Omega \setminus T$ est non vide. Le lemme 2.29 montre que T sépare x de $\Omega \setminus T$, ce qui contredit le fait que Ω et x soient inclus dans $\mathcal{P}(\Gamma_H(\Omega))$. L'ensemble Ω est donc bien égal à $\mathcal{P}(\Gamma_H(\Omega))$. ■

Chapitre 3

Énumération des séparateurs minimaux

Sommaire

3.1	Le treillis des a, b -séparateurs minimaux	24
3.2	Comment produire des séparateurs minimaux	25
3.3	L'algorithme d'énumération de Berry et col.	28
3.4	Un second algorithme	30

L'étude des triangulations des hyper-graphes et des décompositions arborescentes est étroitement liée à celle des séparateurs minimaux. Ceux-ci apparaissent en effet dans nombre de résultats sur les hyper-graphes triangulés, les triangulations ou le calcul effectif de la largeur arborescente. Nous pouvons citer, par exemple, le théorème 2.22 de Dirac, le théorème 5.3 de Parra et Scheffler ou les résultats de Bouchitté et Todinca [BT01a, BT01b]. Énumérer efficacement les séparateurs minimaux nous conduit à approfondir notre connaissance de ces objets. En outre, ceci constitue la première étape de l'algorithme de calcul de largeur arborescente de Bouchitté et Todinca [BT01a, BT01b]. Cet algorithme est polynomial pour une classe de graphes ayant un nombre polynomial de séparateurs minimaux. Nous montrons aux chapitres 6 et 7 que les séparateurs minimaux jouent aussi un rôle important dans l'étude des décompositions en branches.

Escalante [Esc72] montre que l'ensemble des séparateurs minimaux peut être muni d'une structure de *treillis*⁴. En utilisant cette structure, Kloks et Kratsch [KK98] obtiennent un algorithme d'énumération des séparateurs minimaux dont la complexité en temps est en $O(n^5)$ par séparateur. Cet algorithme ainsi que des versions ultérieures [SL97, KK98] énumèrent les a, b -séparateurs minimaux pour tous les couples de sommets (a, b) . La propriété 2.10 montre qu'un séparateur minimal est un a, b -séparateur minimal pour un nombre important de couples (a, b) . Ces algorithmes calculent donc un séparateur minimal un nombre important de fois. Berry et col. [BBC00] présentent un algorithme qui énumère de façon globale les séparateurs minimaux. Ils obtiennent ainsi une complexité en temps $O(nm)$ par séparateur, ce qui constitue un gain important. Néanmoins, ils notent que leur algorithme peut encore obtenir un même séparateur minimal n fois.

⁴C'est-à-dire d'une structure d'ordre où l'ensemble des éléments plus grands (resp. plus petits) que deux éléments x et y admet un plus petit (resp. plus grand) élément.

Nous commençons ce chapitre en présentant la structure de treillis des a, b -séparateurs minimaux et en montrant comment obtenir certains séparateurs minimaux à partir d'autres [KK98, KK98]. À l'aide de ces processus de production, nous présentons l'algorithme de Berry et col. puis un nouvel algorithme qui calcule l'ensemble des a, b -séparateurs minimaux pour a fixé. La stratégie employée par l'algorithme de Berry consiste à produire des séparateurs minimaux le plus vite possible quitte à les obtenir plusieurs fois ; il confie la détection de ces réapparitions à une structure de données généraliste. À l'inverse, notre algorithme s'attache à produire un séparateur minimal une seule fois quitte à ce que cela prenne plus de temps. Pour cela, il maintient des informations supplémentaires qui lui permettent de détecter et donc d'éviter d'éventuelles réapparitions de séparateurs. De cette façon, il peut être adapté à certaines classes d'hyper-graphes. Nous en déduisons au chapitre 8 un algorithme d'énumération des séparateurs minimaux des hyper-graphes planaires en $O(n)$ par séparateur.

3.1 Le treillis des a, b -séparateurs minimaux

Si nous fixons un ensemble H et deux éléments distincts x et y dans H , l'ensemble des parties de H contenant x mais pas y est un treillis pour la relation d'inclusion : l'inf de deux ensembles est leur intersection et leur sup est leur union. Pour les a, b -séparateurs minimaux, la situation est comparable.

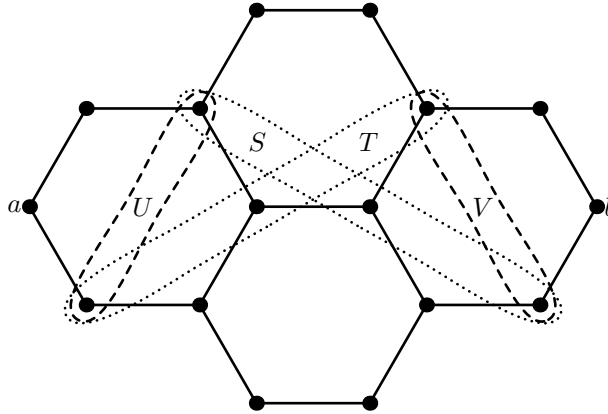
Définition 3.1

Soient S et T deux a, b -séparateurs minimaux d'un hyper-graphe $G = (V, E)$. Le séparateur S est plus proche de a que T (noté $S \prec_a T$) si $C_G^a(S)$ est inclus dans $C_G^a(T)$.

Propriété 3.2 ([KK98]) L'ensemble des a, b -séparateurs minimaux d'un hyper-graphe G forme un treillis pour la relation d'ordre \prec_a .

□ Soient S et T deux a, b -séparateurs minimaux, C_U la composante connexe de $C_G^a(S) \cap C_G^a(T)$ contenant a et U le voisinage dans G de cette composante. Par construction, tout a, b -séparateur minimal plus proche de a que S et T est plus proche de a que U . Nous allons maintenant prouver que U est un a, b -séparateur minimal et donc que l'inf de S et T existe.

Par construction, la composante $C_G^a(U)$ est égale à C_U ; elle est pleine par rapport à U . Montrons que $C_G^b(U)$ l'est aussi. Pour cela, fixons un sommet x de U . Les séparateurs S et T jouant des rôles symétriques, nous pouvons supposer que x appartient à S . La composante $C_G^b(S)$ étant pleine, il existe une chaîne $P_{x \rightarrow b}$ de x à b contenue dans $C_G^b(S) \cup \{b\}$. Les sommets de U sont dans le voisinage de C_U . Par conséquent ce sont soit des sommets de $C_G^a(S)$, soit des sommets de S . La chaîne $P_{x \rightarrow b}$ ne rencontre donc pas l'ensemble



Le séparateur $U = N_G(C_G^a(S \cup T))$ est l'inf de S et T .

De même, $V = N_G(C_G^b(S \cup T))$ est le sup de S et T .

FIG. 3.1 – L'inf et le sup de deux séparateurs minimaux

$U \setminus \{x\}$ ce qui prouve que x est dans le voisinage de $C_G^b(U)$. Le sommet x étant quelconque, la composante connexe $C_G^b(U)$ est pleine. L'ensemble U est donc un a, b -séparateur minimal. L'ensemble des a, b -séparateurs minimaux est un inf-demi-treillis pour la relation \preceq_a .

Si S est plus proche de a que T , par définition, $C_G^a(S)$ est inclus dans $C_G^a(T)$ et donc $C_G^b(T)$ est inclus dans $C_G^b(S)$, c'est-à-dire T est plus proche de b que S . Ceci prouve que $\sup_{\preceq_a}(S, T)$ est égal à $\inf_{\preceq_b}(S, T)$ ce qui achève la démonstration. ■

3.2 Comment produire des séparateurs minimaux

La caractérisation des séparateurs minimaux à l'aide de composantes pleines (prop. 2.10) permet de construire certains a, b -séparateurs minimal à partir de a, b -séparateurs.

Lemme 3.3 *Soit R un a, b -séparateur d'un hyper-graphe G tel que la composante connexe $C_G^a(R)$ soit pleine par rapport à R . L'ensemble $N(C_G^b(R))$ est un a, b -séparateur minimal S .*

□ Comme a et b ne sont pas dans la même composante connexe de $G \setminus R$, toute chaîne de a à b doit sortir de $C_G^b(R)$. Par conséquent, l'ensemble S est un a, b -séparateur. De plus, celui-ci possède au moins une composante pleine : celle contenant b .

Comme S est inclus dans R , la composante connexe $C_G^a(R)$ est incluse dans $C_G^a(S)$. Par conséquent, tout sommet de S est voisin d'un sommet de

$C_G^a(R)$ et donc de $C_G^a(S)$. La composante $C_G^a(S)$ est pleine par rapport à S et donc S est un a, b -séparateur minimal. ■

En utilisant ce lemme, nous pouvons construire facilement certains séparateurs minimaux. Pour cela, il faut partir d'un a, b -séparateur dont la composante contenant a est pleine : c'est le cas du voisinage de a convient. Nous obtenons ainsi la première classe de séparateurs minimaux de Kloks et Kratsch.

Corollaire et définition 3.4 (Séparateurs proches d'un sommet)

Soient G un hyper-graphe et x un sommet de G . Les voisinages dans G des composantes connexes de $G \setminus (N(x) \cup \{x\})$ sont des séparateurs minimaux. Nous disons qu'un tel séparateur est proche de x .

Lemme 3.5 Soient a et b deux sommets non adjacents d'un hyper-graphe G . Le voisinage de la composante connexe de $G \setminus N(a)$ contenant b est le plus petit a, b -séparateur minimal pour la relation \preccurlyeq_a .

□ Soient S le a, b -séparateur minimal inclus dans le voisinage de a et T un autre a, b -séparateur minimal. Aucun sommet de $C_G^b(T)$ n'est adjacent à a sans quoi nous pourrions construire une chaîne de a à b dans l'ensemble $C_G^b(T) \cup \{a\}$. Une telle chaîne éviterait T contredisant le fait que T soit un a, b -séparateur. Nous déduisons que $C_G^b(T)$ est inclus dans $C_G^b(S)$ donc $C_G^a(S)$ est inclus dans $C_G^a(T)$; S est plus proche de a que T . Le séparateur T étant quelconque, S est le plus petit élément du treillis des a, b -séparateurs minimaux. ■

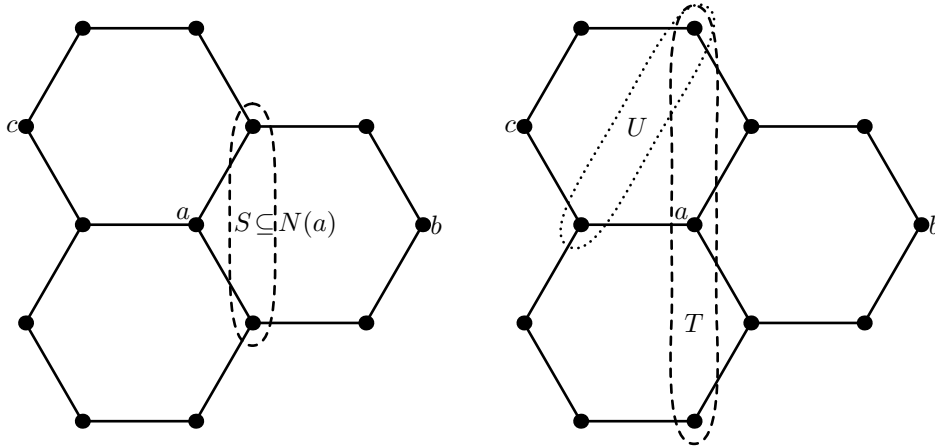
Nous pouvons aussi utiliser le lemme 3.3 pour obtenir de nouveaux séparateurs minimaux à partir d'un séparateur minimal S donné. Pour cela, nous allons « pousser » un sommet de S en dehors du séparateur et ainsi obtenir le second type de séparateurs minimaux de Kloks et Kratsch.

Corollaire et définition 3.6 (Séparateur proche d'un autre)

Soient S un a, b -séparateur minimal d'un hyper-graphe G et x un sommet de S non adjacent à a . Le voisinage $S_{a:x \rightarrow}$ dans G de la composante connexe $C_G^a(S \cup N(x))$ est un séparateur minimal. Le séparateur $S_{a:x \rightarrow}$ est obtenu en poussant x loin de a . Nous dirons qu'il est proche de S .

□ Par construction, $S_{a:x \rightarrow}$ est inclus dans $S \cup N(x)$, $C_G^a(S_{a:x \rightarrow})$ est donc inclus dans $C_G^a(S)$. Ceci implique que $C_G^b(S)$ est inclus dans $C_G^b(S_{a:x \rightarrow})$. Comme de plus, x ne fait pas partie de $S_{a:x \rightarrow}$ et qu'il est voisin d'un sommet de $C_G^b(S)$, le sommet x appartient à $C_G^b(S_{a:x \rightarrow})$. Pour résumer :

- tous les sommets de S sont adjacents à un sommet de $C_G^b(S)$ donc ils sont adjacents à un sommet de $C_G^b(S_{a:x \rightarrow})$;
- tous les sommets de $N(x)$ sont adjacents à x donc ils sont adjacents à un sommet de $C_G^b(S_{a:x \rightarrow})$.



Le séparateur $S = N_G(C_G^b(N(a)))$ est proche de a .
 Le séparateur $U = N_G(C_G^c(S \cup N(a)))$ est proche de T .

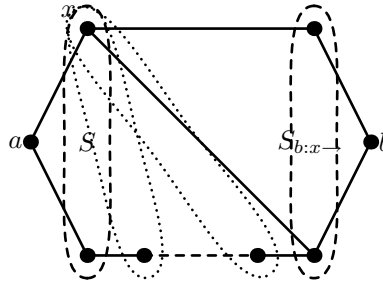
FIG. 3.2 – Séparateurs proches d'un sommet ou d'un autre séparateur

La composante connexe $C_G^b(S_{a:x \rightarrow})$ est pleine. Comme, par construction, $C_G^a(S_{a:x \rightarrow})$ l'est aussi et qu'elles sont distinctes, $S_{a:x \rightarrow}$ est bien un séparateur minimal. ■

Lemme 3.7 Soient S un a, b -séparateur minimal d'un hyper-graphe G et x dans $S \setminus N(b)$. Le séparateur $S_{b:x \rightarrow}$ obtenu en poussant x loin de b est le plus petit a, b -séparateur minimal plus grand que S et ne contenant pas x pour la relation \preceq_a .

□ Soient $S_{b:x \rightarrow}$ le séparateur minimal obtenu en poussant x loin de b et T un autre a, b -séparateur minimal plus grand que S et ne contenant pas x . Aucun sommet de $C_G^b(T)$ n'est adjacent à S ni à $N(x)$ sans quoi nous pourrions construire une chaîne de a à b dans $C_G^a(S) \cup \{x\} \cup C_G^b(T)$. Une telle chaîne éviterait T contredisant le fait que T soit un a, b -séparateur. De ceci nous déduisons que $C_G^b(T)$ est inclus dans $C_G^b(S_{b:x \rightarrow})$ et donc que $S_{b:x \rightarrow}$ est plus proche de a que T . Le séparateur T étant quelconque, $S_{b:x \rightarrow}$ est le plus petit élément du treillis des a, b -séparateurs minimaux plus grand que S et ne contenant pas x . ■

Intuitivement, le séparateur $S_{b:x \rightarrow}$ ne devrait pas être très éloigné de S dans le treillis des a, b -séparateurs minimaux ; le lemme 3.7 va d'ailleurs dans ce sens. Malheureusement, comme l'illustre la figure 3.3, le séparateur $S_{b:x \rightarrow}$ peut être arbitrairement éloigné de S dans ce treillis.



Le séparateur $S_{b:x\rightarrow}$ est obtenu en poussant x loin de b . Il est donc proche de S . Pourtant, dans le treillis des a, b -séparateurs minimaux, tous les séparateurs représentés en pointillés sont entre S et $S_{b:x\rightarrow}$. Le séparateur $S_{b:x\rightarrow}$ peut donc être arbitrairement loin de S .

FIG. 3.3 – Si proche et pourtant si loin.

3.3 L’algorithme d’énumération de Berry et col.

Si nous cherchons à énumérer les séparateurs minimaux d’un hypergraphe et que nous disposons d’un séparateur minimal S , nous pouvons essayer de pousser un sommet x de S dans toutes les directions pour obtenir d’autres séparateurs minimaux. Les séparateurs ainsi obtenus sont les voisinages des composantes connexes de $G \setminus (S \cup N(x))$. En systématisant ce procédé à tous les sommets de S et à chaque nouveau séparateur obtenu, nous obtenons l’algorithme 3.1 [BBC00] dû à Berry, Bordat et Cogis.

Les corollaires 3.4 et 3.6 certifient que cet algorithme calcule bien des séparateurs minimaux. La propriété 3.8 prouve la correction de cet algorithme.

Propriété 3.8 *L’algorithme enumeration calcule tous les séparateurs minimaux d’un hyper-graphe donné.*

□ Supposons par l’absurde que l’algorithme `enumeration` ne produit pas un a, b -séparateur minimal S . Dans la phase d’initialisation, `enumeration` calcule tous les séparateurs minimaux inclus dans les voisinages des sommets de l’hyper-graphe. En particulier, il calcule le a, b -séparateur minimal inclus dans le voisinage de a . L’algorithme calcule donc au moins un a, b -séparateur minimal. De plus, d’après le lemme 3.5, celui-ci est plus proche de a que S .

Parmi les a, b -séparateurs minimaux plus proches de a que S produits par l’algorithme, considérons S_c le a, b -séparateur minimal le plus loin de a possible. Si S_c et S sont distincts, il existe un sommet x de $S_c \setminus S$ appartenant à la composante connexe $C_G^a(S)$. Si nous poussons le sommet x de b vers a , nous obtenons un nouveau séparateur minimal plus grand que S_c . Or d’après le lemme 3.7, celui-ci est plus petit que S ce qui contredit la maximalité de

Algorithme 3.1 enumeration**entrée :** $G = (V, E)$ un hyper-graphe**sortie :** \mathcal{S} : l'ensemble des séparateurs minimaux de G **début** $\mathcal{S} \leftarrow \emptyset$ $\mathcal{L} \leftarrow \emptyset$ // liste des séparateurs non traités**pour chaque** $x \in V$ et $C \in \mathcal{C}_G(x \cup N(x))$ **faire** **si** $N(C) \notin \mathcal{S}$ **alors** $\mathcal{L} \leftarrow \mathcal{L} \cup \{N(C)\}$ $\mathcal{S} \leftarrow \mathcal{S} \cup \{N(C)\}$ **tant que** $\mathcal{L} \neq \emptyset$ **faire** soit $S \in \mathcal{L}$ $\mathcal{L} \leftarrow \mathcal{L} \setminus \{S\}$. **pour chaque** $x \in S$ et $C \in \mathcal{C}_G(S \cup N(x))$ **faire** **si** $N(C) \notin \mathcal{S}$ **alors** $\mathcal{L} \leftarrow \mathcal{L} \cup \{N(C)\}$ $\mathcal{S} \leftarrow \mathcal{S} \cup \{N(C)\}$ **rendre** \mathcal{S} **fin**

S_c . L'algorithme a donc bien calculé le séparateur minimal S . ■

Propriété 3.9 *L'algorithme enumeration peut être implémenté de façon à avoir une complexité en temps $O(nm)$ par séparateur.*

□ Pour atteindre la complexité annoncée, nous avons besoin, pour les ensembles \mathcal{L} et \mathcal{S} , de structures de données permettant des insertions, des suppressions et des tests d'appartenance à un ensemble S en temps $O(|S|)$. Pour plus de précisions sur ce type de structures de données (en particulier les arbres AVL et les files), nous renvoyons le lecteur à [BBC92] ou à [CLR01].

Soit S un séparateur minimal testé dans la boucle de production. Pour chaque sommet x de S , nous calculons le voisinage dans G des composantes connexes de $G \setminus (S \cup N(x))$; chacun de ces calculs de voisinage peut se faire en faisant un parcours de G . Cette étape coûte au total $O(nm)$. Pour chaque séparateur S' ainsi trouvé, il faut tester si S' appartient déjà à \mathcal{S} , ce qui coûte $|S'|$ puis, le cas échéant, l'insérer dans \mathcal{S} et \mathcal{L} ce qui coûte aussi $|S'|$.

Chaque séparateur S' trouvé en poussant le sommet x est le voisinage d'une composante connexe $C_{S'}$ de $G \setminus (S \cup N(x))$. Par conséquent, à chaque sommet de S' nous pouvons associer une hyper-arête qui le relie à $C_{S'}$. En faisant cela pour chaque sommet, il apparaît que la somme des tailles des séparateurs minimaux obtenus en poussant x est au plus m ; le coût induit par la poussée sur x est $O(m)$. L'algorithme effectue au plus n poussées par séparateur ce qui coûte $O(nm)$ par séparateur. ■

Remarque 3.10 Dans le cas des graphes planaires, l'algorithme enumeration fonctionne en $O(n^2)$ par séparateur. En effet, comme le degré moyen d'un sommet d'un graphe planaire est strictement inférieur à six, le nombre d'arêtes d'un tel graphe est inférieur à trois fois le nombre de sommets⁵.

3.4 Un second algorithme

Nous allons présenter un second algorithme général qui calcule un sous-ensemble des séparateurs minimaux d'un hyper-graphe. Avant de le présenter, nous allons définir quelques classes de séparateurs minimaux.

Définitions 3.11

Soient $G = (V, E)$ un hyper-graphe, a un sommet de V et B inclus dans V . L'ensemble $\mathcal{S}_{a,B}$ désigne l'ensemble des a, b -séparateurs minimaux avec

⁵Plus précisément, le degré moyen est au plus égal à $6 - \frac{12}{n}$ et le nombre maximal d'arêtes est $3n - 6$. Pour une démonstration de ces résultats, nous renvoyons le lecteur à [Die00].

b dans B . L'ensemble $\mathcal{S}_{a,V}$ est noté \mathcal{S}_a . Les éléments de \mathcal{S}_a sont appelés des $a, *$ -séparateurs minimaux. Nous ordonnons \mathcal{S}_a par la relation \preccurlyeq_a . Si S est un $a, *$ -séparateur minimal, nous notons B_S l'ensemble des sommets b de V tels que S soit un a, b -séparateur minimal. Soit x dans S et O inclus dans S . L'ensemble $\mathcal{S}_{a,B}(S, O)$ désigne le sous-ensemble de $\mathcal{S}_{a,B}$ constitué des séparateurs contenant O plus éloignés de a que S et $\mathcal{S}_{a,B}^x(S, O)$ le sous-ensemble de $\mathcal{S}_{a,B}(S, O)$ constitué des séparateurs minimaux ne contenant pas x .

La notation B_S est légèrement ambiguë. Il serait plus juste de noter cet ensemble B_S^a mais il n'y aura jamais de doute sur le sommet a auquel la notation fait implicitement référence.

D'après la propriété 2.10, si S est un a, b -séparateur minimal, S est un a, c -séparateur minimal pour n'importe quel sommet c de $C_G^b(S)$; $C_G^b(S)$ est donc inclus dans B_S . Cependant, B_S n'est pas forcément réduit à une unique composante connexe de $G \setminus S$: c'est la réunion des composantes connexes pleines de $G \setminus S$ ne contenant pas a .

L'algorithme que nous allons présenter énumère les séparateurs minimaux de l'ensemble \mathcal{S}_a . Pour cela, il le décompose récursivement en ensembles de la forme $\mathcal{S}_{a,B}(S, O)$ en utilisant la propriété suivante.

Propriété 3.12 Soient S un $a, *$ -séparateur minimal, x dans S , O inclus dans V et $(S_i)_{i \in I}$ les voisinages dans G des composantes connexes de $B_S \setminus N(x)$. Les ensembles S_i sont des $a, *$ -séparateurs minimaux et, en notant \sqcup l'union disjointe :

$$\begin{aligned}\mathcal{S}_{a,B_S}(S, O) &= \mathcal{S}_{a,B_S}(S, O \cup \{x\}) \sqcup \mathcal{S}_{a,B_S}^x(S, O) \\ \mathcal{S}_{a,B_S}^x(S, O) &= \bigsqcup_{i \in I} \mathcal{S}_{a,B_{S_i}}(S_i, O).\end{aligned}$$

□ La première égalité est claire. En effet, si S' est un séparateur minimal de $\mathcal{S}_{a,B_S}(S, O)$, soit x est dans S' auquel cas S' appartient à $\mathcal{S}_{a,B_S}(S, O \cup \{x\})$, soit x n'appartient pas à S' auquel cas S' appartient à $\mathcal{S}_{a,B_S}^x(S, O)$.

L'ensemble $\mathcal{T} = \bigsqcup_{i \in I} \mathcal{S}_{a,B_{S_i}}(S_i, O)$ est un sous-ensemble de $\mathcal{S}_{a,B_S}^x(S, O)$. Le lemme 3.7 justifie l'inclusion réciproque. En effet, soit T un a, b -séparateur minimal de $\mathcal{S}_{a,B_S}^x(S, O)$. D'après le lemme 3.7, le voisinage de la composante connexe de $G \setminus (S \cup N(x))$ contenant b est le a, b -séparateur minimal ne contenant pas x le plus proche de a parmi ceux qui sont plus loin de a que S . Ce séparateur minimal est l'un des S_i . Par conséquent, T appartient à l'un des ensembles $\mathcal{S}_{a,B_{S_i}}(S_i, O)$. Les ensembles $\mathcal{S}_{a,B_S}^x(S, O)$ et \mathcal{T} sont donc égaux.

Prouvons que l'union $\bigsqcup_{i \in I} \mathcal{S}_{a,B_{S_i}}(S_i, O)$ est disjointe. Pour cela, supposons qu'il existe un séparateur minimal S' dans $\mathcal{S}_{a,B_{S_i}}(S_i, O) \cap \mathcal{S}_{a,B_{S_j}}(S_j, O)$. Comme $B_{S'}$ est inclus dans B_{S_i} et dans B_{S_j} , les ensembles B_{S_i} et B_{S_j} ont au

moins un sommet b en commun. Soit C la composante connexe de $B_s \cap N(x)$ contenant b . Cette composante connexe est pleine à la fois par rapport à S_i et par rapport à S_j donc $S_i = N(C) = S_j$ ce qui est absurde. Les ensembles $\mathcal{S}_{a, B_{S_i}}(S_i, O)$ et $\mathcal{S}_{a, B_{S_j}}(S_j, O)$ sont donc disjoints. ■

Algorithme 3.2 `calc_a*_aux` : Énumération de $\mathcal{S}_{a, B_S}(S, O)$

entrée :

$G = (V, E)$ un hyper-graphe
 $a \in V$
 S un $a, *$ -séparateur minimal
 $O \subseteq S$

sortie :

$\mathcal{S}_{a, B_S}(S, O)$

début

$\mathcal{S} \leftarrow \emptyset$

pour chaque $x \in S \setminus O$ **faire**

si $\mathcal{S}_{a, B_S}^x(S, O) \neq \emptyset$ **alors**

pour chaque S_i élément minimal de $\mathcal{S}_{a, B_S}^x(S, O)$ **faire**

$\mathcal{S} \leftarrow \mathcal{S} \cup \text{calc_a*_aux}(G, a, S_i, O)$

$O \leftarrow O \cup \{x\}$

rendre $\mathcal{S} \cup \{S\}$

fin

Propriété 3.13 *Les algorithmes `calc_a*_aux` et `calc_a*` sont corrects.*

□ Étudions tout d'abord l'algorithme `calc_a*_aux`.

La boucle **pour chaque** peut s'écrire sous une forme récursive :

Soit $x \in S \setminus O$

si $\mathcal{S}_{a, B_S}^x(S, O) \neq \emptyset$ **alors**

pour chaque S_i élément minimal de $\mathcal{S}_{a, B_S}^x(S, O)$ **faire**

$\mathcal{S} \leftarrow \mathcal{S} \cup \text{calc_a*_aux}(G, a, S_i, O)$

$\mathcal{S} \leftarrow \mathcal{S} \cup \text{calc_a*_aux}(G, a, S, O \cup \{x\})$

rendre \mathcal{S}

Sous cette forme, la boucle devient une transcription exacte de la propriété 3.12 qui assure que si les séparateurs minimaux S_i sont les voisinages dans G des composantes connexes de $B_S \setminus N(x)$, nous avons les égalités sui-

vantes :

$$\begin{aligned}\mathcal{S}_{a,B_S}(S,O) &= \mathcal{S}_{a,B_S}(S,O \cup \{x\}) \bigsqcup \mathcal{S}_{a,B_S}^x(S,O) \\ \mathcal{S}_{a,B_S}^x(S,O) &= \bigsqcup_{i \in I} \mathcal{S}_{a,B_{S_i}}(S_i,O).\end{aligned}$$

En ce qui concerne l'algorithme `calc_a*`, le lemme 3.5 justifie que les séparateurs minimaux S_i inclus dans $N(a)$ sont les éléments minimaux pour la relation \preceq_a et comme tout a , $*$ -séparateur minimal S est plus éloigné de a que l'un des S_i , S appartient à l'un des ensembles $\mathcal{S}_{a,B_{S_i}}(S_i, \emptyset)$. ■

Algorithme 3.3 `calc_a*` : Énumération de l'ensemble \mathcal{S}_a

entrée :

$G = (V, E)$: un hyper-graphe
 $a \in V$

sortie :

\mathcal{S}_a : l'ensemble des a , $*$ -séparateurs minimaux de G

début

$\mathcal{S} \leftarrow \emptyset$

pour chaque S séparateur minimal inclus dans $N(a)$ **faire**

$\mathcal{S} \leftarrow \mathcal{S} \cup \text{calc_a*_aux}(G, a, S, \emptyset)$

rendre \mathcal{S}

fin

Propriété 3.14 *Les algorithmes `calc_a*_aux` et `calc_a*` peuvent être implémentés de façon à avoir une complexité en temps $O(nm)$ par séparateur.*

□

Considérons tout d'abord l'algorithme `calc_a*_aux`.

- Un élément minimal S_i de $\mathcal{S}_{a,B_S}^x(S,O)$ est le voisinage dans G d'une composante connexe de $B_S \setminus N(x)$. De plus, S_i doit contenir O . Par conséquent, ces séparateurs peuvent être obtenus grâce à un parcours de G . Chacune de ces exécutions coûte donc $O(m)$;
- Pour tester si l'ensemble $\mathcal{S}_{a,B_S}^x(S,O)$ est vide ou non, nous pouvons naïvement calculer ses éléments minimaux. Ce faisant, nous sommes ramenés au problème précédent qui se résoud donc en $O(m)$;
- Si l'ensemble \mathcal{S} est implémenté grâce à une liste, chaque insertion dans \mathcal{S} se fait en temps constant.

Chaque passage dans la boucle coûte donc $O(m)$. Il y a $|S \setminus O|$ passages dans cette boucle donc au plus n passages. Le temps de calcul pour un séparateur minimal S donné est donc $O(nm)$.

En ce qui concerne l'algorithme `calc_a*`, le lemme 3.5 justifie que les séparateurs minimaux inclus dans $N(a)$ sont les voisinages des composantes connexes de $G \setminus N(a)$. Ils peuvent donc être calculés grâce à un parcours de l'hyper-graphe. Comme l'algorithme `calc_a*_aux` peut être implémenté de façon à avoir une complexité en temps $O(nm)$ par séparateur, nous obtenons une complexité totale en $O(nm)$ par séparateur pour `calc_a*`. ■

Nous revenons sur ce second algorithme au chapitre 8 et nous montrons comment, dans le cas des hyper-graphes planaires 3-connexes, le spécialiser pour de façon à ce qu'il fonctionne en $O(n)$ par séparateur. Dans ce cas particulier, notre algorithme est plus efficace que l'algorithme `enumeration`.

Chapitre 4 Diviser pour régner

Sommaire

4.1	Décompositions arborescentes	35
4.2	Décompositions en branches	40
4.3	Matriochkas et arbres de matriochkas	43
4.3.1	Matriochkas	44
4.3.2	Arbres de matriochkas	45
4.3.3	Extensions d'arbres de matriochkas	47
4.4	Quelques liens entre ces deux décompositions	50

Le principe d'un schéma « diviser pour régner » est de diviser un problème de façon récursive en sous-problèmes, de résoudre chacun des sous-problèmes puis de fusionner les résultats partiels pour obtenir une solution générale. Un exemple d'une telle technique est le « tri fusion » : pour trier une liste l , la découper en deux listes l_1 et l_2 de tailles équivalentes ; trier chacune des deux listes puis fusionner les deux sous-listes triées.

Pour qu'une telle technique soit efficace, il faut s'arranger pour que les sous-problèmes se résolvent bien de façon inductive et pour que l'opération de fusion se fasse bien. Ainsi, la façon dont le problème principal est découpé est très importante. Dans le cas du tri fusion, si la liste est découpée de façon totalement déséquilibrée en formant une liste de taille 1 et une seconde de taille $n - 1$, nous obtenons le tri par insertion dont la complexité en moyenne est $O(n^2)$ alors qu'elle est $O(n \ln(n))$ pour le tri fusion⁶.

4.1 Décompositions arborescentes

Définition 4.1 (Décomposition arborescente)

Soit $G = (V, E)$ un hyper-graphe. Une décomposition arborescente T_θ de G est un arbre T muni d'une fonction θ d'étiquetage des nœuds de T de telle sorte que :

- (i) les étiquettes de T soient des ensembles de sommets de G ;
- (ii) chaque sommet de G apparaît dans au moins une étiquette ;
- (iii) les extrémités de chaque hyper-arête de G apparaissent simultanément dans au moins une étiquette de T ;

⁶Pour une présentation plus précise de ces algorithmes de tri, voir [BBC92] ou [CLR01].

- (iv) si un sommet apparaît dans les étiquettes de deux nœuds u et v de T , alors il apparaît dans les étiquettes de tous les nœuds de la chaîne de T de u à v .

La figure 4.2 est une décomposition arborescente du graphe G de la figure 4.1.

Cette définition ressemble beaucoup à celle d'arbre des cliques (déf. 2.24). Effectivement, si T_θ est une décomposition arborescente d'un hyper-graphe G et que nous ajoutons à G des arêtes de telle sorte que chacune des étiquettes de T_θ induise une clique de G , le sur-hyper-graphe H ainsi défini est une triangulation de G . Un certain nombre de problèmes comme celui de la clique maximale ou de la coloration se résolvent bien sur les hyper-graphes triangulés. Il semble naturel qu'une « bonne » décomposition arborescente et donc une « bonne » triangulation de G puisse aider à résoudre un problème sur G . Cependant, il n'est pas clair que cette intuition se confirme de façon générale. Nous allons montrer comment certaines approches « diviser pour régner » sur des graphes mènent à des décompositions arborescentes.

Afin d'utiliser ce type de techniques, il faut choisir comment « découper » le graphe en morceaux plus petits tout en conservant une trace de la structure de G pour pouvoir combiner les solutions particulières en une solution globale.

Une approche possible est celle de l'algorithme 4.1 ; elle consiste à utiliser un a, b -séparateur minimal S de G . Ce séparateur minimal induit plusieurs composantes connexes $(C_i)_{i \in I}$ de $G \setminus S$. Pour construire des « morceaux », nous allons successivement porter notre attention sur chacune des composantes C_i en supprimant les autres. Si la composante C_i n'est pas pleine, nous supprimons aussi les sommets de S qui ne sont pas voisins d'un sommet de C_i . Néanmoins, nous complétons S en une clique pour garder la trace du fait que deux sommets quelconques de S sont reliés par une chaîne dont l'intérieur est dans une composante pleine de S . Nous définissons ainsi le graphe G_i correspondant au sous-graphe $G[C_i \cup N(C_i)]$ dans lequel nous avons complété $N(C_i)$ en une clique.

Pour obtenir une décomposition complète de G , nous utilisons récursivement la même méthode sur les graphes $(G_i)_{i \in I}$. La figure 4.1 illustre un tel découpage récursif.

Compléter les sommets de S en une clique de G_a et de G_b présente un autre intérêt. En effet, les sommets de S ne peuvent pas être séparés lors des découpages successifs de G_a et de G_b . Ils apparaissent donc dans au moins une des cliques irréductibles que nous obtenons à l'issue de ces découpages successifs. Par conséquent, nous pouvons extraire une structure arborescente représentant l'ordre des découpages :

- si G est une clique, nous définissons un arbre T_θ ne contenant qu'un seul nœud x et donc l'étiquette est V ;

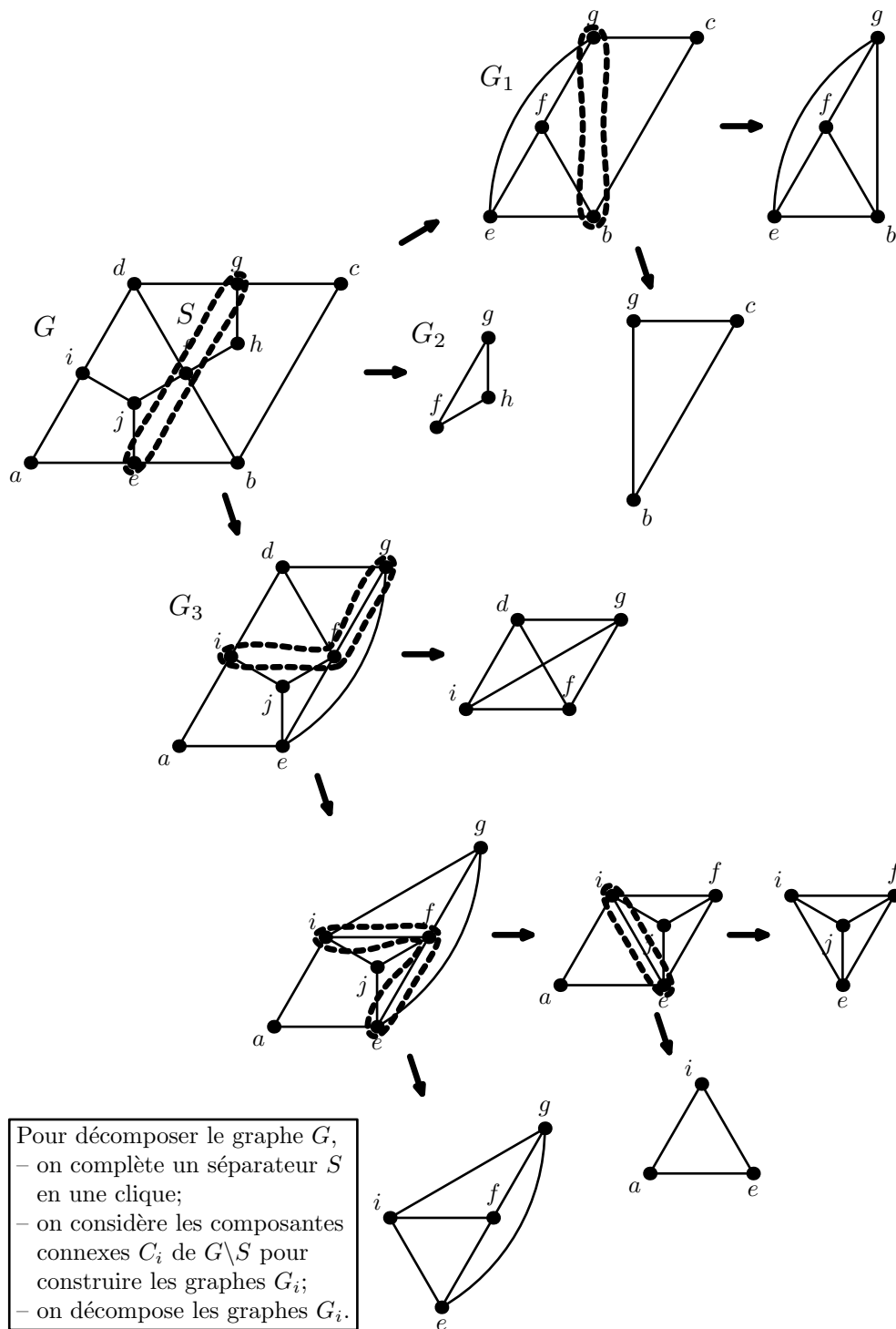


FIG. 4.1 – Une décomposition récursive d'un graphe G

- sinon, nous obtenons récursivement plusieurs décompositions $(T_i)_{\theta_i}$ correspondant aux graphes G_i . La clique S ne pouvant être subdivisée, l'étiquette d'un nœud y_i de T_i contient $N(C_i)$. Nous obtenons alors un arbre T_θ en reliant les nœuds y_i à un nœud x dont l'étiquette est S .

Algorithme 4.1 `decomposition_arborescente`

entrée :

$G = (V, E)$ un graphe connexe.

sortie :

T : une décomposition arborescente de G

début

si G est une clique **alors**

$T \leftarrow$ l'arbre réduit à un nœud contenant V

sinon

$S \leftarrow$ un a, b -séparateur minimal de G

Compléter $G[S]$ en une clique

$T \leftarrow$ l'arbre réduit à un nœud x contenant S

pour chaque composante connexe C_i de $G \setminus S$ **faire**

$T_i \leftarrow$ `decomposition_arborescente`($G[C_i \cup N(C_i)]$)

$y_i \leftarrow$ un nœud de T_i dont l'étiquette contient $N(C_i)$

attacher T_i à T en reliant x et y_i

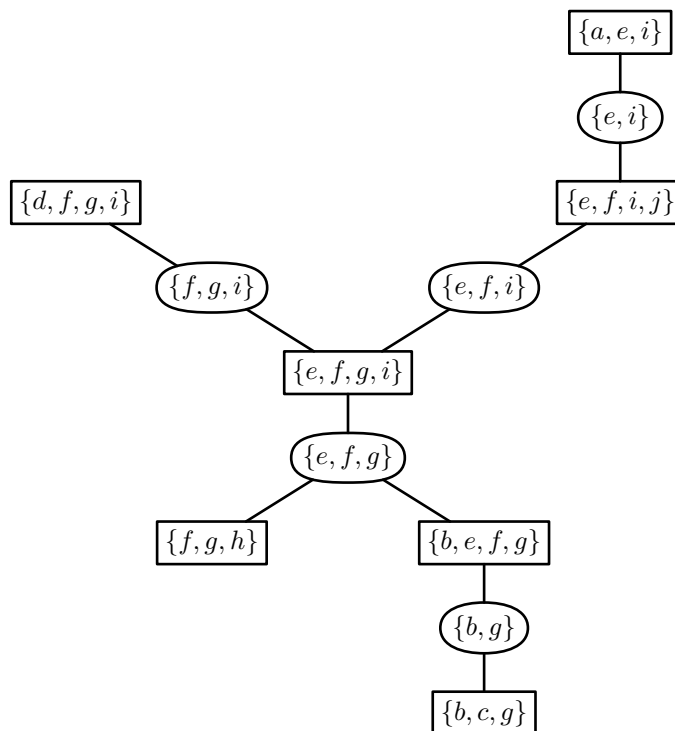
rendre (T)

fin

L'algorithme `decomposition_arborescente` formalise le processus que nous venons de décrire. Il est aisé de se convaincre qu'il produit bien des décompositions arborescentes.

Nous avons vu comment certaines décompositions arborescentes peuvent correspondre à des schémas généraux « diviser pour régner » sur des graphes. Pour évaluer l'efficacité d'un tel schéma, il faut prendre en compte deux paramètres. D'une part, la taille des séparateurs minimaux conditionne l'opération de fusion des solutions partielles ; nous devons donc chercher à utiliser des séparateurs minimaux les plus petits possibles. D'autre part, la taille des cliques irréductibles conditionne le calcul des solutions partielles initiales. Celles-ci correspondant aux étiquettes de la décomposition, nous devons donc aussi chercher à minimiser leur taille. Comme les séparateurs minimaux considérés sont tous inclus dans une étiquette de la décomposition, le paramètre naturel à minimiser est la taille d'une plus grosse étiquette.

Plus formellement, nous définissons la largeur arborescente :



Cet arbre correspond à la décomposition de la figure 4.1. Les nœuds elliptiques correspondent aux séparateurs minimaux utilisés et les autres aux cliques irréductibles.

FIG. 4.2 – L'arbre de décomposition de la figure 4.1

Définition 4.2 (Largeur arborescente)

Soit G un hyper-graphe. La largeur d'une décomposition arborescente T_θ de G est la taille d'une plus grosse étiquette de T_θ moins un. La largeur arborescente de G notée $\text{tw}(G)$ ⁷ est la plus petite largeur d'une de ses décompositions arborescentes.

Le « moins un » sert à ce que les arbres et les forêts aient une largeur arborescente de un.

4.2 Décompositions en branches

Dans une optique « diviser pour régner », seules les opérations de découpe et de fusion devraient être coûteuses. Obtenir une solution sur un cas de base devrait se faire rapidement sinon ce cas de base devrait pouvoir se redécouper. De ce point de vue, les décompositions arborescentes ne sont pas optimales. En effet, la taille du plus grand séparateur minimal utilisé pour diviser le graphe dans une telle décomposition peut être arbitrairement plus petite que la largeur de cette décomposition. De ce point de vue, les décompositions en branches sont plus satisfaisantes.

Définitions 4.3 (Décomposition en branches, largeur de branches)

Soit G un hyper-graphe. Une décomposition en branches T_τ de G est un arbre T muni d'une fonction τ d'étiquetage des feuilles de T de telle sorte que :

- (i) les nœuds internes de T soient de degré trois ;
- (ii) la fonction τ soit une bijection de l'ensemble des feuilles de T vers les hyper-arêtes de G .

À chaque arête e de T correspond une séparation $\mathcal{E} = \{E_1, E_2\}$ où E_1 et E_2 sont les ensembles d'arêtes étiquetant les feuilles des deux composantes connexes de $T \setminus \{e\}$. La frontière de l'arête e est celle de la séparation \mathcal{E} . La largeur de la décomposition notée $\text{bw}(T)$ est la plus grosse taille d'une frontière de ses arêtes et la largeur de branches de G notée $\text{bw}(G)$ ⁸ est la plus petite largeur d'une décomposition de G .

La figure 4.4 donne un graphe et une de ses décompositions en branches.

Pour « découper » le graphe G , les décompositions en branches utilisent une partition $\{E_1, E_2\}$ de l'ensemble de ses arêtes. Nous obtenons alors deux graphes $G[E_1]$ et $G[E_2]$. Mais, comme ce mode de « découpage » permet de diviser des cliques, nous ne pouvons plus compléter la frontière entre ces deux graphes en une clique pour respecter la structure du graphe initial. Nous sommes donc amenés à considérer les deux hyper-graphes contractés $G_{/E_2}$ et $G_{/E_1}$.

⁷Le terme anglo-saxon est *treewidth* ce qui explique la notation $\text{tw}(G)$.

⁸Le terme anglo-saxon est *branchwidth* ce qui explique la notation $\text{bw}(G)$.

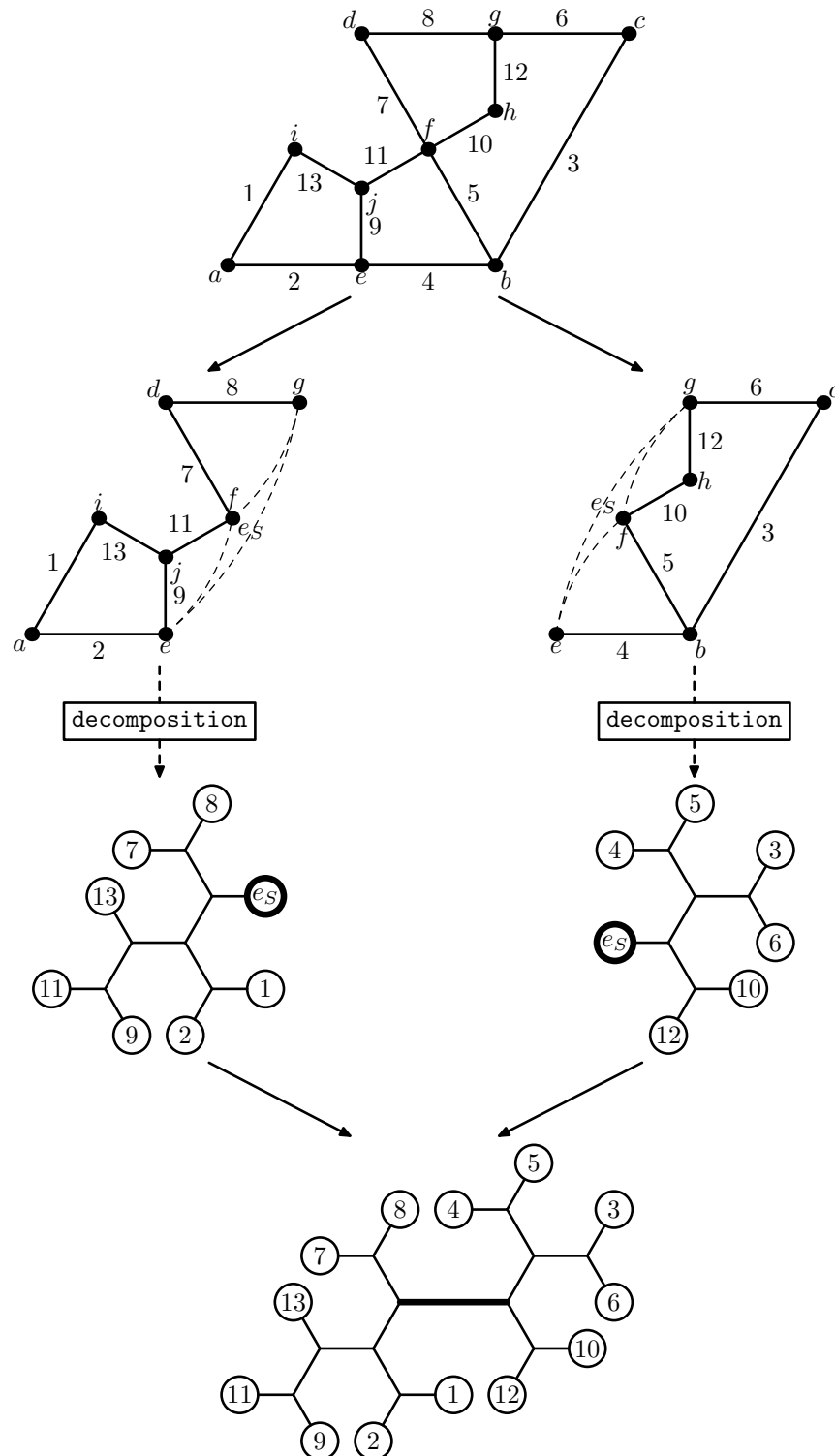
Définition 4.4 (Hyper-graphe contracté)

Soient $G = (V, E)$ un hyper-graphe et E' un ensemble d'hyper-arêtes de G . L'hyper-graphe $G[E \setminus E']$ auquel nous avons rajouté l'hyper-arête $\partial(E')$ est noté G/E' . C'est l'hyper-graphe G dans lequel les hyper-arêtes de E' sont contractées.

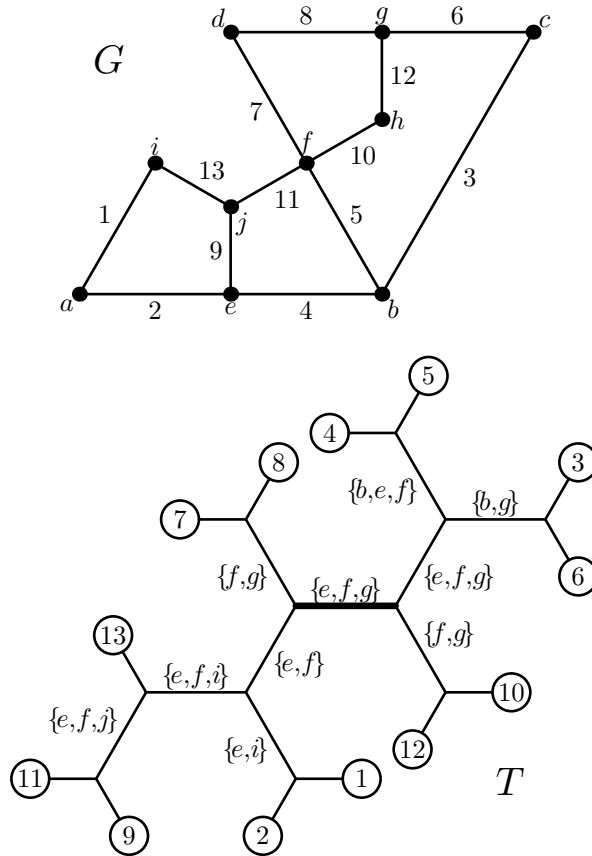
Autant les décompositions arborescentes peuvent être présentées en se restreignant aux graphes, autant le bon cadre pour les décompositions en branches est celui des hyper-graphes.

L'algorithme `decomposition_branches` (algo. 4.2) construit un tel schéma de décomposition. Il est aisé de se convaincre que ces schémas sont bien des décompositions en branches et que toutes les décompositions en branches d'un hyper-graphe peuvent être obtenues ainsi. La figure 4.3 illustre une de ses exécutions. Les arêtes de la décomposition correspondent aux séparations choisies pour décomposer G .

Algorithme 4.2 `decomposition_branches`**entrée :** $G = (V, E)$ un hyper-graphe connexe**sortie :** T : une décomposition en branches**début****si** l'hyper-graphe a 1 hyper-arête **alors** $T \leftarrow$ l'arbre réduit à un nœud contenant l'hyper-arête de G **si** l'hyper-graphe a 2 hyper-arêtes **alors** $T \leftarrow$ l'arbre réduit à une arête dont les feuilles contiennent les hyper-arêtes de G **si** l'hyper-graphe a 3 hyper-arêtes **alors** $T \leftarrow$ l'étoile à 3 branches dont les feuilles contiennent les hyper-arêtes de G **sinon**Soit $\{E_1, E_2\}$ une séparation de G avec $|E_1| \geq 2$ et $|E_2| \geq 2$ $S \leftarrow \partial(E_1)$ $V_1 \leftarrow \{v \in V \mid v \text{ est incident à au moins une arête de } E_1\}$ $T_1 \leftarrow \text{decomposition_branches}(G_1 = (V_1, E_1 \cup \{S\}))$ $V_2 \leftarrow \{v \in V \mid v \text{ est incident à au moins une arête de } E_2\}$ $T_2 \leftarrow \text{decomposition_branches}(G_2 = (V_2, E_2 \cup \{S\}))$ $T \leftarrow T_1$ et T_2 recollés sur la feuille contenant S **rendre** (T)**fin**

FIG. 4.3 – Une exécution de l'algorithme `decomposition_branches`

Certaines décompositions en branches peuvent donc être vues comme l'application d'une technique « diviser pour régner » sur des hyper-graphes. L'efficacité d'un tel schéma dépend de la taille d'une plus grosse frontière utilisée lors de la décomposition. La largeur de branches permet bien de juger de la qualité d'une décomposition en branches (voir fig. 4.4).



L'arbre T est une décomposition en branches du graphe G . L'arête de T représentée en gras correspond à la séparation $\{E_1, E_2\} = (\{1, 2, 7, 8, 9, 11, 13\}, \{3, 4, 5, 6, 10, 12\})$. Sa frontière est $\{e, f, g\}$ et sa taille est donc de trois. Les autres ensembles donnés correspondent aux frontières des arêtes à côté desquelles ils sont représentés. La largeur de T est trois.

FIG. 4.4 – Exemple de décomposition en branches

4.3 Matriochkas et arbres de matriochkas

L'algorithme 4.2 que nous venons de présenter est proche de l'algorithme 4.1 mais celui-ci permet de poursuivre les décompositions là où le premier

s'arrête et ainsi de décomposer plus d'hyper-graphes. De plus comme nous l'avons déjà noté, la taille d'un plus grand séparateur minimal utilisé pour définir une décomposition arborescente est toujours plus petite que la taille d'une des cliques de base. Il semble donc naturel que ce soit aussi le cas si nous décomposons plus avant ces cliques de base avec des séparations. Ceci semble suggérer que $\text{bw}(G) \leq \text{tw}(G) + 1$ et donc que la largeur de branche est une *obstruction* à la largeur arborescente. C'est d'ailleurs exactement dans ce contexte que Robertson et Seymour [RS91] les ont introduites. Ils obtiennent un résultat plus fort dont nous donnons la démonstration au théorème 4.25 : si G est un hyper-graphe, en rappelant que $\beta(G)$ est la taille maximale d'une hyper-arête de G , alors

$$\max(\text{bw}(G), \beta(G)) \leq \text{tw}(G) + 1 \leq \max(\lceil 3 \text{bw}(G)/2 \rceil, \beta(G)).$$

En utilisant un théorème min/max sur la largeur de branches [RS91] et un résultat profond de topologie plane [RS94] précisant ce premier théorème, Seymour et Thomas [ST94] donnent un algorithme polynomial pour calculer la largeur de branches d'un hyper-graphe planaire. Cet algorithme donne la meilleure approximation de la largeur arborescente des hyper-graphes planaires à ce jour.

Dans le chapitre 6, nous sommes amenés à opérer certaines optimisations locales sur des décompositions en branches. Pour pouvoir les effectuer facilement, nous commençons par étudier en profondeur la structure des décompositions que produit l'algorithme 4.2. Dans un premier temps, nous nous intéressons aux séparations qu'il utilise ce qui nous conduit à définir des *matriochkas*. Dans un second temps, dans le but de mieux cerner la structure arborescente de ces objets, nous introduisons la notion d'*arbre de matriochka* qui généralise celle de décomposition en branches puis nous étudions comment certaines opérations simples sur les matriochkas s'interprètent sur les arbres de matriochkas. Ce dernier point nous permet d'obtenir une version itérative de l'algorithme `decomposition_branches` : nous montrons dans la propriété 4.11 qui existe une correspondance bijective entre matriochkas et arbres de matriochkas.

4.3.1 Matriochkas

A priori, les séparations que produit `decomposition_branches` sont de plusieurs types : en effet, il fait intervenir une séparation de G puis des séparations des graphes obtenus à partir de G en contractant des ensembles d'hyper-arêtes en des hyper-arêtes. Mais si nous observons un arbre obtenu par l'algorithme (fig. 4.4), rien ne différencie les arêtes de l'arbre. Le lemme suivant justifie que les séparations utilisées puissent être vues comme des séparations de G et précise une propriété de l'ensemble de ces séparations.

Lemme 4.5 *Soient $G = (V, E)$ un hyper-graphe, $\{E_1, E_2\}$ une séparation de G . Une séparation de $G_{/E_2}$ induit une séparation de G . De plus, celle-ci ne se chevauche pas $\{E_1, E_2\}$.*

□ Notons S la frontière de la séparation initiale $\{E_1, E_2\}$. Soit $\{E'_1, E'_2\}$ une séparation de $G_{/E_2}$. Nous pouvons supposer que l'hyper-arête S appartient à l'ensemble E'_2 . Si nous « décontractons » l'hyper-arête S dans $\{E'_1, E'_2\}$, nous obtenons la séparation $\{E'_1, (E'_2 \cup E_2) \setminus \{S\}\}$ de G . De plus, cette séparation ne chevauche pas $\{E_1, E_2\}$ car E_2 est disjoint de E'_1 . ■

Dans le lemme 4.5, nous avons vu que les séparations que manipule l'algorithme `decomposition_branches` sont associées à des séparations de G qui ne se chevauchent pas. Ceci nous amène à introduire la notion de matriochka dérivée de celle de carving définie par Robertson et Seymour.

Définitions 4.6 (Matriochka, matriochka parallèle)

Une matriochka de G est la réunion des éléments d'un ensemble de séparations de G qui ne se chevauchent pas deux à deux. Nous notons généralement \mathcal{C} les matriochkas. Une matriochka est parallèle si les frontières de ses éléments sont parallèles deux à deux.

Le lemme 4.5 justifie que `decomposition_branches` manipule des matriochkas. De plus, comme n'importe quelle séparation de $G_{/E_1}$ et $G_{/E_2}$ ne chevauche pas $\{E_1, E_2\}$, cet algorithme construit des matriochkas dont nous montrons (prop 4.16) qu'elles sont complètes, en leur ajoutant des éléments au fur et à mesure. Autrement dit, l'algorithme `decomposition_branches` ne fait qu'étendre la matriochka vide.

Définitions 4.7 (Extension de matriochka, matriochka complète)

Une matriochka \mathcal{C} est plus fine qu'une matriochka \mathcal{C}' si \mathcal{C}' est inclus dans \mathcal{C} . Les matriochkas les plus fines sont complètes.

4.3.2 Arbres de matriochkas

L'algorithme `décomposition_branches` associe à certaines matriochkas des structures arborescentes qui les représentent. La définition de matriochka ne fait pas apparaître clairement de telles structures. Ceci nous amène à introduire la notion d'*arbres de matriochkas* qui étend celle de décomposition en branches.

Définitions 4.8 (Branche)

*Soit $G = (V, E)$ un hyper-graphe. Une branche ${}^*T_\tau$ de G de support X strictement inclus dans E est un arbre enraciné *T muni d'une fonction τ d'étiquetage des feuilles de T de telle sorte que :*

- (i) les étiquettes de T soient des ensembles éventuellement vides d'hyper-arêtes de G ;

- (ii) les étiquettes soient disjointes et de réunion égale à X ;
- (iii) si l'étiquette d'un nœud est vide, ce nœud ait au moins deux fils.

À chaque nœud de $*T$ correspond une sous-branche de $*T_\tau$. Nous notons \mathcal{D}_{*T_τ} l'ensemble des supports des sous-branches de $*T_\tau$.

Définitions 4.9 (Arbre de matriochka)

Soit $G = (V, E)$ un hyper-graphe. Un arbre de matriochka T_τ de G est un arbre T muni d'une fonction τ d'étiquetage des feuilles de T de telle sorte que :

- (i) les étiquettes de T soient des ensembles éventuellement vides d'hyper-arêtes de G ;
- (ii) les étiquettes soient disjointes et de réunion égale à E ;
- (iii) si l'étiquette d'un nœud est vide, ce nœud est de degré au moins trois.

À chaque arête e de T correspondent deux composantes connexes de $T \setminus \{e\}$, ces composantes connexes sont des branches. Leurs supports forment une séparation de G . La frontière de l'arête e est celle de la séparation qui lui est associée. Nous notons \mathcal{C}_{T_τ} l'ensemble des supports de branches de T_τ .

Remarque 4.10 Les décompositions en branches sont des arbres de matriochkas et nous montrons (prop. 4.16) qu'ils correspondent à des matriochkas complètes.

Un arbre de matriochka est donné par deux branches $*T_1^e$ et $*T_2^e$ dont les supports forment une séparation de G et dont les racines sont reliées par l'arête e . De plus, \mathcal{C}_{T_τ} est égal à

$$\mathcal{D}_{*T_1^e} \cup \{E \setminus X \mid X \in \mathcal{D}_{*T_1^e}\} \cup \mathcal{D}_{*T_2^e} \cup \{E \setminus X \mid X \in \mathcal{D}_{*T_2^e}\}.$$

Propriété 4.11 L'ensemble des matriochkas de G est en bijection avec l'ensemble des arbres de matriochkas de G . Nous notons $T_{\mathcal{C}}$ l'arbre de matriochka correspondant à la matriochka \mathcal{C} et $\tau_{\mathcal{C}}$ la fonction d'étiquetage correspondante.

□ Soit T_τ un arbre de matriochka de G . L'ensemble \mathcal{C}_{T_τ} est la réunion des éléments d'une famille de séparations. De plus, si X et Y appartiennent à \mathcal{C}_{T_τ} , X est le support d'une branche $*T^e$ de T_τ et Y celui d'une branche $*T^f$. Quitte à considérer leurs complémentaires, nous pouvons supposer que $*T^e$ ne contient pas l'arête f et que $*T^f$ ne contient pas e . De cette façon, les ensembles X et Y sont disjoints. Les séparations que définit T_τ ne se chevauchent pas et l'ensemble \mathcal{C}_{T_τ} est donc une matriochka.

Réciproquement, soient \mathcal{C} une matriochka et X un élément de \mathcal{C} . Notons \mathcal{D}_X et $\mathcal{D}_{E \setminus X}$ l'ensemble des éléments de \mathcal{C} inclus respectivement dans X et dans $E \setminus X$. L'ensemble \mathcal{C} est égal à

$$\mathcal{D}_X \cup \{E \setminus Y \mid Y \in \mathcal{D}_X\} \cup \mathcal{D}_{E \setminus X} \cup \{E \setminus Y \mid Y \in \mathcal{D}_{E \setminus X}\}.$$

En utilisant l'égalité de la remarque 4.10, nous n'avons qu'à montrer qu'il existe une branche *T_X telle que les ensembles \mathcal{D} et $\mathcal{D}_{{}^*T_X}$ soient égaux. Montrons ceci par induction sur la taille de \mathcal{D}_X .

Si \mathcal{D}_X est réduit au singleton $\{X\}$, la branche réduite à un seul nœud x dont l'étiquette $\tau(x)$ est X convient. Sinon soient X_1, \dots, X_p les éléments de \mathcal{D}_X distincts de X maximaux pour l'inclusion. Les éléments de \mathcal{C} ne se chevauchant pas, les X_i sont disjoints. Pour chaque X_i , l'ensemble des éléments de \mathcal{D}_X inclus dans X_i est égal à \mathcal{D}_{X_i} . Par induction, il existe une branche $({}^*T_i)_{\tau_i}$ telle que \mathcal{D}_{X_i} soit égal à $\mathcal{D}_{({}^*T_i)_{\tau_i}}$. Soit *T_X la branche obtenue en reliant les racines des branches *T_i à une racine x dont l'étiquette $\tau(x)$ est $X \setminus \cup_{i=1}^n X_i$ et en maintenant les étiquettes de chacun des autres nœuds. La branche ainsi obtenue vérifie les conditions voulues, ce qui achève la démonstration. ■

4.3.3 Extensions d'arbres de matriochkas

Les arbres de matriochkas constituent donc une représentation duale des matriochkas. Cependant, nous avons ordonné les matriochkas pour l'inclusion. L'algorithme 4.2 construit certaines matriochkas correspondant à des décompositions en branches en les étendant petit à petit. Nous allons montrer comment interpréter ces opérations d'extension directement sur les arbres de matriochkas. Certaines de ces transformations apparaissent par exemple dans [Hic00] ou dans les travaux de Cook et Seymour [CS02] et sont utilisées dans des heuristiques de construction de décompositions en branches. La transformation que nous considérons ici est plus générale que celles de Hicks. En fait, la propriété 4.15 montre qu'elle est la plus générale possible.

Les figures 4.5, 4.6 et 4.7 illustrent certaines opérations d'extensions élémentaires au sens que les matriochkas correspondantes contiennent exactement les éléments d'une séparation de plus que celles de départ. Nous ne le justifions pas car ces transformations sont des cas particuliers d'une unique transformation plus générale.

Dans chacune des trois transformations des figures 4.5, 4.6 et 4.7, un nœud u est remplacé par deux nœuds v et w reliés, les voisins de u et les éléments de son étiquette sont répartis entre v et w . Pour que l'arbre étiqueté obtenu soit un arbre de matriochka, il faut que la condition iii soit vérifiée ce qui implique que l'étiquette de v soit non vide ou qu'au moins deux voisins de u soient voisins de v . Le nœud w doit vérifier les mêmes conditions. Plus formellement :

Définition 4.12 (Nœud découpable)

Soit T_τ un arbre de matriochka d'un hyper-graphe G . Un nœud u de T est découpable, s'il existe un sous-ensemble non vide I inclus dans $N(u)$ et un sous-ensemble U de $\tau(u)$ tels que

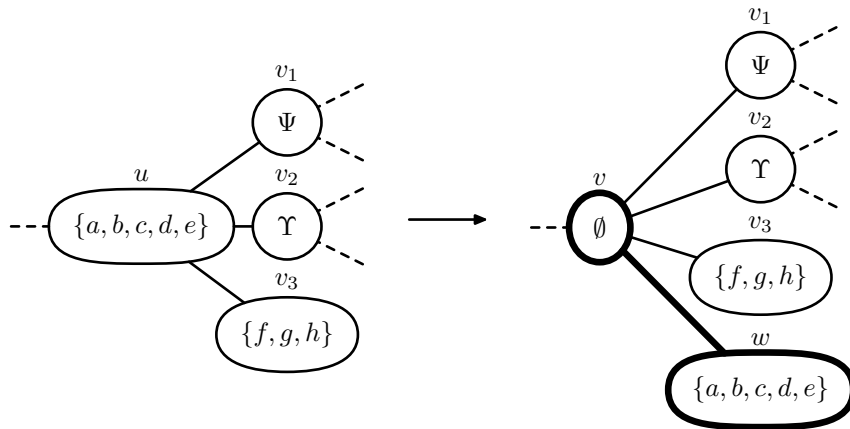


FIG. 4.5 – Un premier type d'extension

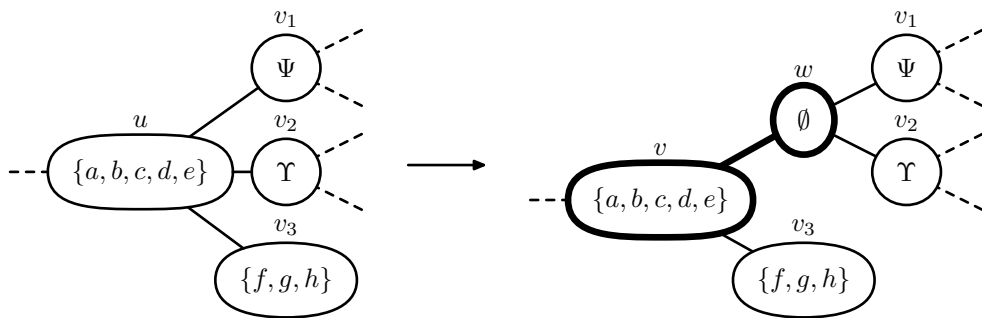


FIG. 4.6 – Un deuxième type d'extension

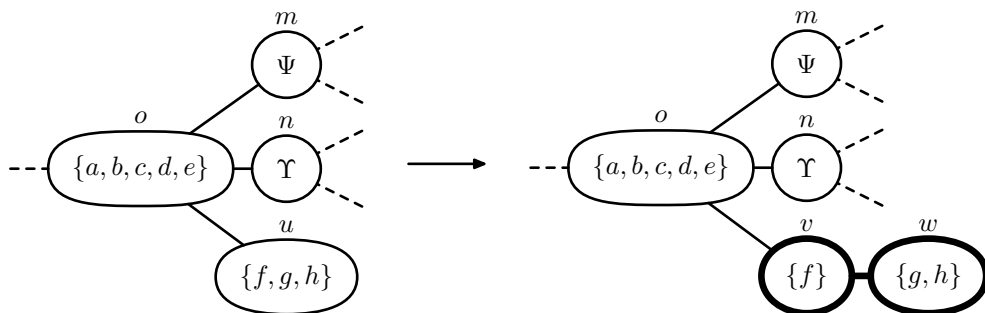


FIG. 4.7 – Un troisième type d'extension

- $|I| \geq 2$ ou $U \neq \emptyset$ et
- $|N(u) \setminus I| \geq 2$ ou $\tau(w) \setminus U \neq \emptyset$.

Les ensembles I et U découpent le nœud w .

Pour qu'un nœud ne soit pas découpable, il faut qu'il ait exactement trois voisins et que son étiquette soit vide ou que ce soit une feuille et que son étiquette soit un singleton.

Par conséquent, un arbre de matriochka possède au moins un nœud découpable si et seulement si ce n'est pas une décomposition en branches.

De plus, chaque arête d'un arbre de matriochka correspond à une séparation de G . En effectuant l'opération décrite ci-dessus, nous ajoutons exactement une arête à l'arbre de matriochka. Par conséquent :

Propriété et définition 4.13

Soient I et U qui découpent un nœud u d'un arbre de matriochka T_τ .

Soit l'arbre étiqueté T'_τ , obtenu à partir de T_τ en remplaçant le nœud u par deux nouveaux nœuds reliés v et w , en reliant v (resp. w) aux nœuds de I (resp. $N(u) \setminus I$) et en choisissant $\tau'(v)$ et $\tau'(w)$ respectivement égaux à U et à $\tau(u) \setminus U$. L'arbre T'_τ est un arbre de matriochka étendant T_τ tel que $\mathcal{C}_{T'_\tau}$ contienne les éléments d'exactly une séparation de plus que \mathcal{C}_{T_τ} . Il est obtenu en découpant le nœud u selon I et U .

Nous pouvons maintenant donner une version itérative de l'algorithme 4.2.

Algorithme 4.3 decomposition_iteratif

entrée :

$G = (V, E)$ un hyper-graphe connexe

sortie :

T : une décomposition en branches

début

$T \leftarrow$ un nœud x dont l'étiquette est $\tau(x) = V$

tant que $\exists I$ et U découpant un nœud u de T **faire**

 découper u selon I et U

rendre (T)

fin

Lemme 4.14 Soit \mathcal{C}' strictement plus fine que \mathcal{C} . Il existe une matriochka \mathcal{C}'' telle que :

- $\mathcal{C} \subset \mathcal{C}'' \subseteq \mathcal{C}'$;
- $T_{\mathcal{C}''}$ est obtenu en découpant un nœud de $T_{\mathcal{C}}$.

□ Soit X dans $\mathcal{C}' \setminus \mathcal{C}$. Quitte à considérer son complémentaire, nous pouvons supposer que X est inclus dans un élément de \mathcal{C} . Soit Y dans \mathcal{C} contenant X minimal pour l'inclusion et $*T_Y$ la branche de T_τ de racine y dont le support est Y . Le choix de Y fait que le support d'une sous-branche stricte de T_Y est soit inclus dans X , soit disjoint de X . Soient I l'ensemble des fils z_i de y tels que le support de la sous-branche de $*T_Y$ de racine z_i soit inclus dans X et U l'ensemble $\tau(y) \cap X$. Comme X est strictement inclus dans Y , soit I ne contient pas tous les fils de Y , soit U n'est pas égal à $\tau(y)$. Les ensembles I et U découpent le nœud y . La matriochka obtenue en découpant y selon I et U étend \mathcal{C} et contient X . Elle vérifie donc les hypothèses voulues. ■

La propriété suivante est un corollaire immédiat du lemme 4.14.

Propriété 4.15 *Soit \mathcal{C}' plus fine que \mathcal{C} . Nous pouvons obtenir \mathcal{C}' à partir de \mathcal{C} en découpant successivement des nœuds de $T_{\mathcal{C}}$.*

Pour construire un arbre de matriochka, nous pouvons donc partir d'une décomposition triviale et l'étendre progressivement. De plus, nous sommes maintenant en mesure de montrer que les décompositions en branches correspondent aux matriochkas complètes.

Propriété 4.16 *Les décompositions en branches sont les arbres de matriochkas correspondant aux matriochkas complètes.*

□ Soient T_τ une décomposition en branches et \mathcal{C}_{T_τ} la matriochka correspondante. Si \mathcal{C}_{T_τ} n'est pas complète, d'après le lemme 4.14, nous pouvons découper un nœud de T_τ pour l'étendre. Or aucun nœud d'une décomposition en branches n'est découppable. La matriochka \mathcal{C}_{T_τ} est donc complète.

Inversement si T_τ est un arbre de matriochka correspondant à une matriochka complète, il est impossible de découper un nœud de T_τ . L'arbre T_τ est donc ternaire et ses nœuds internes ont des étiquettes vides. C'est une décomposition en branches. ■

4.4 Quelques liens entre ces deux décompositions

Nous avons introduit les décompositions arborescentes et les décompositions en branches en suivant une démarche similaire. Nous montrons maintenant certaines propriétés qui justifient que cette similarité ne s'arrête pas là.

Tout d'abord, ces deux paramètres sont compatibles avec la relation de minoration.

Propriété 4.17 *Soient H un sur-hyper-graphe d'un hyper-graphe G . Nous avons : $\text{tw}(G) \leq \text{tw}(H)$.*

□ Soit T_{θ_H} une décomposition arborescente de H dont la largeur est $\text{tw}(H)$. Si nous ne gardons dans les étiquettes de T_{θ_H} que les sommets de G , nous obtenons une décomposition arborescente T_{θ_G} dont la largeur est inférieure ou égale à celle de H . Par conséquent,

$$\text{tw}(G) \leq \text{tw}(T_{\theta_G}) \leq \text{bw}(T_{\theta_H}) = \text{bw}(H). \quad \blacksquare$$

Propriété 4.18 *Soit H un sur-hyper-graphe d'un hyper-graphe G . Nous avons : $\text{bw}(G) \leq \text{bw}(H)$.*

□ Soit $(T_H)_{\tau_H}$ une décomposition en branches de H dont la largeur est $\text{bw}(H)$. Si nous effeuillons T_H en lui retirant les feuilles qui ne correspondent pas à des hyper-arêtes de G , nous obtenons un arbre qui contient des nœuds de degré deux. Nous pouvons supprimer de tels nœuds en reliant directement par une arête leurs deux voisins. En itérant ce processus, nous obtenons une décomposition en branches $(T_G)_{\tau_G}$ de G . De plus, par construction, la frontière de chaque arête de $(T_G)_{\tau_G}$ est incluse dans la frontière de l'arête correspondante dans $(T_H)_{\tau_H}$. Par conséquent,

$$\text{bw}(G) \leq \text{bw}((T_G)_{\tau_G}) \leq \text{bw}((T_H)_{\tau_H}) = \text{bw}(H). \quad \blacksquare$$

Les propriétés que nous abordons maintenant constituent le cœur de cette thèse et presque tous les résultats en sont issus.

Nous montrons comment associer à une décomposition en branches T_τ une famille de sous-arbres de T de telle sorte que le nombre de sous-arbres contenant une arête e de T donne la taille de cette arête. Cette famille de sous-arbres permet aussi, en s'intéressant aux sous-arbres passant par les nœuds de T de définir une décomposition arborescente. De cette façon, à partir d'un objet arborescent commun T , nous obtenons à la fois une décomposition en branches en nous intéressant aux arêtes de T et une décomposition arborescente en nous intéressant à ses nœuds. De plus, les largeurs de branches et aborescente peuvent être atteintes sur cet objet.

Robertson et Seymour [RS91] utilisent cette dualité entre décomposition arborescente et décomposition en branches sans l'expliciter pour obtenir le théorème min/max 4.25.

Lemme 4.19 *Soient T_τ une décomposition en branches d'un hyper-graphe G et v un nœud interne de T . Tout sommet appartenant à la frontière d'une arête incidente à v appartient à la frontière d'au moins deux arêtes incidentes à v .*

□ L'étiquette $\tau(v)$ est vide. Par conséquent, si le sommet x appartient à la frontière de l'arête e_1 incidente à v , il existe une arête de G incidente à x dans au moins deux branches de $T \setminus \{v\}$. Le sommet x appartient donc à la frontière d'au moins deux arêtes incidentes à v . ■

Propriété 4.20 (voir fig. 4.8) Soient T_τ une décomposition en branches d'un hyper-graphe $G = (V, E)$, x un sommet de G et F_x l'ensemble des feuilles de T correspondant aux hyper-arêtes de G incidentes à x .

L'ensemble des arêtes de T_τ dont l'étiquette contient le sommet x induit le sous-arbre T_x de T dont les feuilles sont les nœuds de F_x .

□ Soient v_1 et v_2 dans F_x et e une arête de T appartenant au chemin de v_1 à v_2 . Les deux nœuds v_1 et v_2 ne sont pas dans la même composante connexe de $T \setminus e$. Par conséquent, le support de chacune des branches de $T \setminus e$ contient une hyper-arête incidente à x . Le sommet x appartient donc à l'étiquette de l'arête e . Ceci prouve que les étiquettes des arêtes du chemin de v_1 à v_2 contiennent toutes x . Les nœuds v_1 et v_2 étant distincts, c'est aussi le cas pour les arêtes du sous-arbre T_x .

Considérons maintenant une arête e n'appartenant pas à T_x . Puisque e n'est sur aucun chemin reliant deux feuilles de F_x , toutes les feuilles de F_x se trouvent dans une même branche de $T \setminus e$ et par conséquent le support de l'autre branche ne contient aucune hyper-arête incidente à x . La frontière de e ne contient pas le sommet x . ■

Définition 4.21 (Largeur arborescente d'une matriochka)

Soient T_τ un arbre de matriochka d'un hyper-graphe G et v un nœud de T . Posons $\theta_\tau(v)$ l'ensemble des sommets x de G tels que le sous-arbre T_x de T passe par v ou tels que x soit incident à une arête de τ_v . L'arbre T_{θ_τ} est la décomposition arborescente induite par T_τ .

La largeur arborescente d'une matriochka ou d'un arbre de matriochka est la largeur arborescente de la décomposition arborescente induite.

Notation 4.22

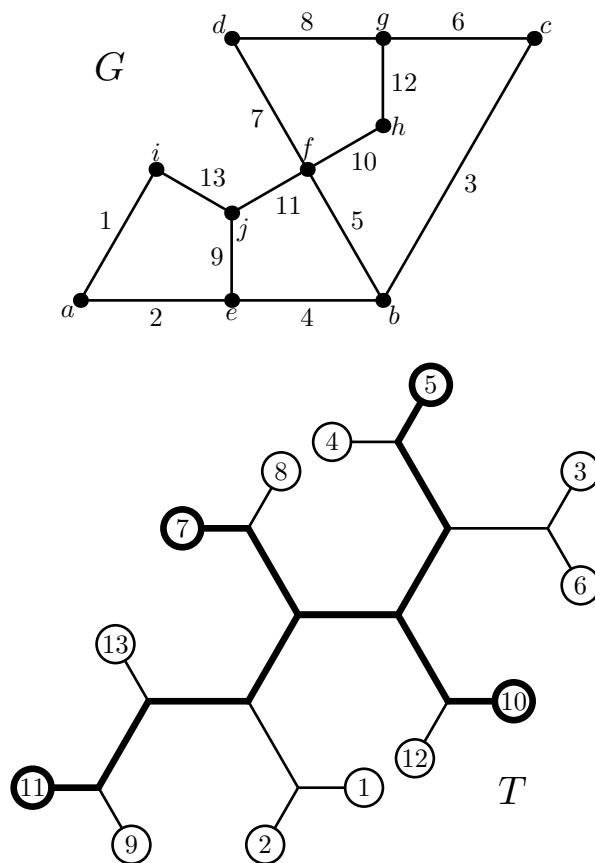
Nous rappelons que les notations T_τ et T_θ font chacune référence à deux objets : un arbre et une fonction d'étiquetage. Ainsi, T_τ et T_θ sont deux décompositions utilisant le même arbre. De même, la décomposition T_{θ_τ} que nous venons de définir désigne une décomposition arborescente utilisant le même arbre que la décomposition en branches T_τ et dont la fonction d'étiquetage θ_τ est définie à partir de τ .

Nous montrons aussi comment définir une décomposition en branches T'_{τ_θ} à partir d'une décomposition arborescente T_θ . Nous pouvons donc comparer les décompositions arborescentes T'_{θ_τ} et T_θ .

Le théorème suivant, bien que simple, est très important. Il justifie que nous puissions nous intéresser uniquement aux décompositions en branches.

Théorème 4.23

Si l'arbre de matriochka T'_τ étend l'arbre T_τ , alors sa largeur arborescente est inférieure ou égale à celle de T_τ .



Dans l'arbre de matriochka T du graphe G , les arêtes en gras sont celles dont la frontière contient le sommet f .

FIG. 4.8 – Le sous-arbre induit par un sommet.

□ La propriété 4.15 permet de supposer que l'arbre T'_τ est obtenu en découpant un nœud de T_τ .

Soient w le nœud de T découpé en u et v pour obtenir T' et x un sommet de G . Le sous-arbre induit par les nœuds s de T' dont les étiquettes $\theta_\tau(s)$ contiennent x est le sous-arbre de T' reliant les nœuds de T' dont les étiquettes contiennent une arête de G incidente à x . Par conséquent, comme les branches de $T' \setminus \{u, v\}$ sont les branches de $T \setminus \{w\}$ et que les étiquettes $\tau(u)$ et $\tau(v)$ partitionnent $\tau(w)$, les étiquettes $\theta_\tau(s)$ et $\theta_{\tau'}(s)$ des nœuds communs à T et T' sont les mêmes. Si le sommet x appartient à l'étiquette du nœud u ou v dans T'_τ , alors il appartient à celle de w dans T_τ . La largeur arborescente de T'_τ est donc inférieure ou égale à celle de T_τ . ■

Réciproquement,

Propriété et définition 4.24

Soit T_θ une décomposition arborescente d'un hyper-graphe G . Il existe un arbre de matriochka T_{τ_θ} tel que la largeur arborescente de T_{τ_θ} soit inférieure ou égale à celle de T_θ . L'arbre de matriochka T_{τ_θ} est dérivé de T_θ .

□ Pour chaque hyper-arête e de G , il existe un nœud u de T dont l'étiquette $\theta(u)$ contient les extrémités de e . Plaçons donc l'hyper-arête e dans l'étiquette $\tau_\theta(u)$. L'arbre T_{τ_θ} est un arbre de matriochka. De plus, par construction, le sous-arbre T_v associé au sommet v est un sous-arbre du sous-arbre de T dont les étiquettes contiennent v . La largeur arborescente de T_{τ_θ} est donc inférieure ou égale à celle de T_θ . ■

Nous sommes maintenant en mesure de prouver le théorème suivant. Rappelons que $\beta(G)$ désigne la taille d'une plus grosse hyper-arête de G .

Théorème 4.25 ([RS91])

Soit $G = (V, E)$ un hyper-graphe,

$$\max(\text{bw}(G), \beta(G)) \leq \text{tw}(G) + 1 \leq \max(\lfloor 3 \text{bw}(G)/2 \rfloor, \beta(G)).$$

□ Soit T_τ une décomposition en branches quelconque de G .

La largeur arborescente de la décomposition T_{θ_τ} induite par T_τ est supérieure ou égale à celle de G . Or les feuilles de T correspondent aux hyper-arêtes de G . La taille de la plus grosse feuille de T_{θ_τ} est donc $\beta(G)$. Si v est un nœud interne de T_{θ_τ} et si S_1, S_2 et S_3 sont les trois frontières correspondantes aux arêtes incidentes à v dans T , l'étiquette de v dans T_{θ_τ} est $\Omega = S_1 \cup S_2 \cup S_3$. De plus, d'après le lemme 4.19, chaque sommet de Ω apparaît dans au moins deux frontières S_i distinctes. Donc $2|\Omega| \leq |S_1| + |S_2| + |S_3| \leq 3 \text{bw}(T_\tau)$. Au final, comme la décomposition T_τ est quelconque, nous avons bien :

$$\text{tw}(G) + 1 \leq \max(\lfloor 3 \text{bw}(G)/2 \rfloor, \beta(G)).$$

Inversement, soient T_θ une décomposition arborescente et T_{τ_θ} un arbre de matriochka dérivé. D'après la propriété 4.24, la largeur arborescente de

T_{τ_θ} est inférieure ou égale à celle de T_θ . En étendant T_{τ_θ} , nous obtenons une décomposition en branches $T'_{\tau'}$. D'après le théorème 4.23, la largeur arborescente de $T'_{\tau'}$ est inférieure ou égale à celle de T_{τ_θ} . Comme chaque frontière de $T'_{\tau'}$ est incluse dans l'étiquette d'un nœud de $T_{\theta_{\tau'}}$, nous avons $\text{bw}(T'_{\tau'}) \leq \text{tw}(T'_{\theta_{\tau'}}) \leq \text{tw}(T_\theta)$. De plus, les extrémités d'une hyper-arête apparaissant dans au moins une étiquette de T_θ et la décomposition T_θ étant quelconque, nous avons bien :

$$\max(\text{bw}(G), \beta(G)) \leq \text{tw}(G) + 1. \quad \blacksquare$$

Le fait que nous puissions associer une décomposition arborescente à une décomposition en branches a une autre conséquence importante :

Propriété 4.26 *La largeur de branches d'un hyper-graphe est la plus petite largeur de branches d'une de ses triangulations.*

□ Si un hyper-graphe possède au plus deux hyper-arêtes, il est triangulé et ne possède qu'une décomposition en branches. La triangulation induite par cette unique décomposition en branches est l'hyper-graphe lui-même. La propriété est donc vraie dans le cas des hyper-graphes ayant une ou deux hyper-arêtes.

Supposons que G soit un hyper-graphe ayant au moins trois hyper-arêtes et considérons T_τ une décomposition en branches de largeur minimale de G et H la triangulation induite par T_τ .

Si G est un sous-hyper-graphe strict de H , il existe deux sommets x et y reliés dans H mais pas dans G . Ces deux sommets appartiennent à une clique maximale Ω de H . Cette clique est induite par l'étiquette $\theta_\tau(u)$ d'un nœud u de T . Les étiquettes des feuilles de T_τ correspondant à des hyper-arêtes, le nœud u est un nœud interne de T . D'après le lemme 4.19, les sommets x et y apparaissent chacun dans au moins deux frontières associées à des arêtes incidentes à u . Il existe donc une arête e de T_τ dont la frontière contient les deux sommets x et y . Soient $*T_1$ et $*T_2$ les deux branches de $T \setminus \{e\}$. Construisons une décomposition en branches $T'_{\tau'}$, en plaçant un nouveau nœud v d'étiquette vide entre $*T_1$ et $*T_2$ et en ajoutant à v une feuille w dont l'étiquette $\tau'(w)$ est l'arête xy . La décomposition $T'_{\tau'}$, ainsi obtenue est une décomposition en branches de l'hyper-graphe G' égal à $(V, E \cup \{xy\})$. De plus, l'arête xy appartenant à la frontière de e , la largeur de branches de G' est la même que celle de G .

En répétant ce procédé, nous construisons une décomposition en branches de H dont la largeur est celle de T_τ . La largeur de branches de H est inférieure ou égale à celle de G . La propriété 4.18 permet de conclure en montrant que les largeurs de branches de G et de H sont les mêmes. \blacksquare

Cette propriété est l'analogie de la propriété suivante pour la largeur arborescente.

Propriété 4.27 *La largeur arborescente d'un hyper-graphe est la plus petite largeur arborescente d'une de ses triangulations.*

□ Soit T_θ est une décomposition arborescente d'un hyper-graphe G . Considérons H le sur-hyper-graphe de G obtenu en ajoutant à G des arêtes de telle sorte que chaque étiquette de la décomposition T_θ induise une clique. Par construction, T_θ est aussi une décomposition arborescente de H . Donc $\text{tw}(H)$ est inférieur ou égal à $\text{tw}(G)$. D'autre part, si H est un sur-hyper-graphe de G , toute décomposition arborescente de H est aussi une décomposition arborescente de G . En combinant ces deux résultats, nous avons bien l'égalité voulue. ■

De plus, la largeur arborescente d'un hyper-graphe triangulé H est la taille maximale d'une clique de H moins un et celle-ci se calcule en temps linéaire.

Sommaire

5.1	Triangulations serrées	57
5.2	Familles complètes de blocs d'un hyper-graphe	60

La notion de décomposition arborescente est très liée à celle de triangulation. Les propriétés 4.26 et 4.27 énoncent que la largeur de branches et la largeur arborescente d'un hyper-graphe sont égales aux plus petites largeurs correspondantes d'une triangulation de l'hyper-graphe.

Les hyper-graphes triangulés étant structurellement simples, nous pouvons penser que le calcul de leur largeur de branches est plus aisé que pour un hyper-graphe quelconque ; c'est en particulier le cas pour la largeur arborescente. Il peut donc être intéressant de chercher à calculer la largeur arborescente ou la largeur de branches de triangulations de G pour en déduire celle de G . Cependant, en utilisant cette approche, nous perdons d'un côté ce que nous gagnons de l'autre : au lieu de calculer un paramètre pour un seul hyper-graphe, nous devons le calculer pour un grand nombre de triangulations. Nous cherchons donc à restreindre la classe des triangulations à considérer.

Dans ce chapitre, nous introduisons une classe de triangulations des hyper-graphes : les *triangulations serrées* et nous adaptons un algorithme de Bouchitté et Todinca [BT01a] qui prend en entrée une famille de blocs (déf 2.33), qui décide s'il existe une triangulation serrée n'utilisant que des blocs de cette famille et qui construit, si elle existe, une telle triangulation. Nous montrons au chapitre 6 comment utiliser cet algorithme pour calculer la largeur arborescente d'un hyper-graphe d'une part et la largeur de branches d'un hyper-graphe d'autre part.

5.1 Triangulations serrées

Pour construire des triangulations, nous pouvons utiliser le théorème de caractérisation des hyper-graphes triangulés 2.22 : un hyper-graphe est triangulé si et seulement si tous ses séparateurs minimaux induisent des cliques. Celui-ci suggère l'algorithme suivant : tant que G n'est pas triangulé, choisir un séparateur minimal S qui n'induit pas de clique de G et compléter S en une clique de G .

Nous obtenons ainsi une famille croissante d'hyper-graphes dont le dernier élément est triangulé. De plus, un séparateur minimal S' d'un hyper-graphe complété est un séparateur minimal de l'hyper-graphe initial et

comme S' ne peut séparer les sommets de S , les séparateurs minimaux de l'hyper-graphe complété sont parallèles à S . La triangulation finale est obtenue en complétant des séparateurs minimaux deux à deux parallèles de G .

Notation 5.1

Soit Γ une famille de séparateurs minimaux deux à deux parallèles d'un hyper-graphe G . L'hyper-graphe G_Γ est l'hyper-graphe G auquel nous avons ajouté des arêtes de sorte que chaque élément de Γ induise une clique de G_Γ .

Définitions 5.2 (Triangulations minimales)

Une triangulation minimale d'un hyper-graphe est une triangulation minimale pour l'inclusion.

Une clique maximale potentielle d'un hyper-graphe G est un ensemble de sommets de G qui induit une clique d'une triangulation minimale de G .

Les triangulations obtenues par le processus ci-dessus sont des triangulations minimales. De plus, toutes les triangulations minimales sont de cette forme comme l'énonce le théorème de Para et Scheffler.

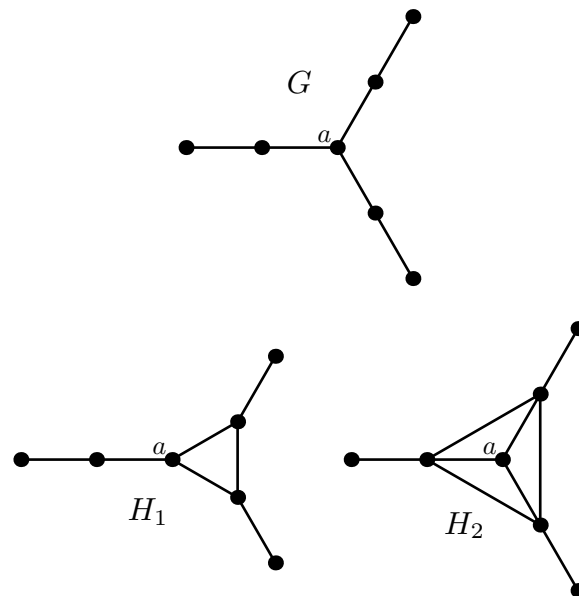
Théorème 5.3 ([PS97])

Soit Γ une famille de séparateurs minimaux deux à deux parallèles de G maximale pour l'inclusion. L'hyper-graphe G_Γ est une triangulation minimale de G .

Soit H une triangulation minimale d'un hyper-graphe G . L'ensemble Δ_H est une famille de séparateurs minimaux deux à deux parallèles de G maximale pour l'inclusion et H est égale à G_{Δ_H} .

Une méthode plus générale pour construire des triangulations consiste à utiliser la propriété 2.23 et le théorème 2.35. Ceux-ci montrent que les séparateurs minimaux d'un hyper-graphe triangulé H sont deux à deux parallèles et que les cliques maximales de H forment des blocs minimaux. Ainsi, les cliques maximales de H sont entièrement caractérisées par les séparateurs minimaux de H . Pour construire une triangulation d'un hyper-graphe G , il suffit de se donner une famille \mathcal{F} de séparateurs minimaux deux à deux parallèles et de compléter les blocs minimaux que forment les séparateurs de \mathcal{F} en des cliques. Si la famille \mathcal{F} est maximale pour l'inclusion, nous retrouvons les triangulations minimales.

Pour caractériser les triangulations que produit ce second processus, nous pouvons remarquer que les séparateurs minimaux d'une telle triangulation sont des séparateurs minimaux de G . Cependant, comme l'illustre la figure 5.1, ce n'est pas suffisant. Par construction, les cliques maximales d'une telle triangulation sont des blocs de G et donc les séparateurs minimaux de H doivent induire les mêmes composantes connexes dans G et dans H .



Les séparateurs minimaux des graphes H_1 et H_2 sont des séparateurs minimaux du graphe triangulé G .

Le séparateur $\{a\}$ n'induit pas les mêmes composantes connexes dans G et dans H_1 . Le graphe H_1 n'est pas une triangulation serrée de G alors que le sur-graphe H_2 de H_1 en est une.

FIG. 5.1 – Un bloc de H n'induit pas forcément un bloc de G .

Définition 5.4 (Triangulation serrée)

Une triangulation H d'un hyper-graphe G est serrée si les séparateurs minimaux de H sont des séparateurs minimaux de G et si les séparateurs minimaux de H induisent les mêmes composantes connexes de H et de G .

Nous nous intéressons uniquement à des triangulations serrées dans la suite.

5.2 Familles complètes de blocs d'un hyper-graphe

Nous nous intéressons au problème suivant : étant donnée une famille de blocs \mathcal{F} , existe-t-il une triangulation dont les blocs appartiennent à \mathcal{F} ? Une telle triangulation n'existe pas toujours mais plus nous avons de blocs, plus il y a de chances que ce soit le cas. En contrepartie, plus la famille initiale est grande, plus la construction d'une éventuelle triangulation prend de temps.

Définitions 5.5 (Famille complète de blocs)

Si \mathcal{F} est une famille de blocs d'un hyper-graphe G , nous disons d'un séparateur minimal qu'il borde \mathcal{F} s'il borde un bloc de \mathcal{F} . Nous notons $\Gamma_{\mathcal{F}}$ l'ensemble $\bigcup_{\Omega \in \mathcal{F}} \Gamma_{\Omega}$ des séparateurs minimaux qui bordent \mathcal{F} ⁹.

Une famille \mathcal{F} de blocs est complète si pour tout séparateur S de $\Gamma_{\mathcal{F}}$ et pour toute composante connexe pleine C de $G \setminus S$, il existe un bloc Ω de \mathcal{F} bordé par S et inclus dans $C \cup S$.

Une famille complète peut être vide mais si ce n'est pas le cas, nous montrons que nous pouvons en extraire une triangulation de G . En particulier, si H est une triangulation serrée de G , l'ensemble des cliques maximales de H forme une famille complète de blocs de G .

Lemme 5.6 *L'ensemble des cliques maximales d'un hyper-graphe triangulé H forme une famille complète de blocs de H .*

L'ensemble des cliques maximales d'une triangulation serrée H d'un hyper-graphe G forme une famille complète de blocs de G .

□ D'après le théorème 2.35, toute clique maximale de H est un bloc de H . Soit \mathcal{F} la famille des cliques maximales de H .

Soit S un séparateur minimal de H . D'après le théorème 2.22, S est une clique de H donc S est inclus dans une clique maximale Ω de H . D'après le lemme 2.34, les ensembles $\Delta_H(\Omega)$ et $\Gamma_H(\Omega)$ sont égaux. Le séparateur S borde Ω . Ceci prouve que l'ensemble $\Gamma_{\mathcal{F}}$ des séparateurs qui bordent \mathcal{F} est Δ_H .

Soit S un séparateur minimal de H et C une composante connexe pleine dans $\mathcal{C}_H^*(S)$. L'hyper-graphe H étant triangulé, l'hyper-graphe H' égal à $H[S \cup C]$ l'est aussi. Dans H' , S est une clique qui induit une composante

⁹ Γ_{Ω} désigne l'ensemble des séparateurs minimaux qui bordent Ω

pleine donc d'après la propriété 2.30, S n'induit pas de clique maximale. Par induction, le lemme est vrai. Soit Ω une clique maximale de H' bordée par S . La clique Ω est aussi une clique maximale de H et le séparateur minimal S borde aussi Ω dans H .

Si H est une triangulation serrée de G , par définition, les séparateurs minimaux de H sont des séparateurs minimaux de G et ils induisent les mêmes composantes connexes dans G et dans H . Comme de plus, les seuls séparateurs minimaux de G qui bordent les cliques maximales de H sont les séparateurs minimaux de H , l'ensemble des cliques maximales de H forme une famille complète de blocs de G . ■

À partir d'une famille de blocs \mathcal{F} , nous cherchons à construire une famille complète \mathcal{F}_c incluse dans \mathcal{F} . Pour cela, nous pouvons procéder par élimination. En effet, si un bloc Ω de \mathcal{F} appartient à une famille complète, par définition, pour chaque séparateur S le bordant et chaque composante pleine C de $\mathcal{C}_G^*(S)$, il existe un bloc Ω' de \mathcal{F} bordé par S et inclus dans $C \cup S$. S'il n'existe pas de tel bloc Ω' , nous pouvons éliminer Ω car il n'appartient à aucune famille complète incluse dans \mathcal{F} . Ainsi, tous les blocs qui restent à l'issue de ce processus appartiennent à une famille complète incluse dans \mathcal{F} . En fait, l'ensemble de bloc obtenu est lui-même une famille complète ; c'est la plus grande famille complète incluse dans \mathcal{F} .

L'algorithme 5.1 formalise ce processus.

Algorithme 5.1 *elimination*

entrée :

\mathcal{F} : une famille de blocs d'un hyper-graphe G

$\Gamma_{\mathcal{F}}$: l'ensemble des séparateurs qui bordent \mathcal{F}

sortie :

\mathcal{F}_S : la plus grande famille complète incluse dans \mathcal{F}

début

tant que $\exists S \in \Gamma_{\mathcal{F}}$ et $C \in \mathcal{C}_G^*(S)$ tels que

$\forall \Omega \in \mathcal{F}, S \notin \Gamma_{\Omega}$ ou $\Omega \not\subseteq C \cup S$ **faire**

$\mathcal{F} \leftarrow \mathcal{F} \setminus \{\Omega \mid S \in \Gamma_{\Omega}\}$

mettre à jour $\Gamma_{\mathcal{F}} = \bigcup_{\Omega \in \mathcal{F}} \Gamma_{\Omega}$

rendre \mathcal{F}

fin

Propriété 5.7 *L'algorithme 5.1 est correct.*

À partir d'une famille complète \mathcal{F} de blocs d'un hyper-graphe G , l'algorithme 5.3 extrait un arbre des cliques d'une triangulation de G dont les cliques maximales sont des blocs de \mathcal{F} .

Algorithme 5.2 `extraction_aux`

entrée : G : un hyper-graphe S : un séparateur minimal de G ou \emptyset $C \in C_G^*(S)$: une composante pleine de S \mathcal{F} : une famille complète de G Γ : l'ensemble des séparateurs qui bordent \mathcal{F} **sortie :** \mathcal{T} : un arbre des cliques enraciné d'une triangulation H de $G[C \cup S]$ dont les cliques maximales sont des blocs de \mathcal{F} et dont la racine contient S .**début****si** $S = \emptyset$ **alors** Soit $\Omega \in \mathcal{F}$ avec $\Omega \subseteq C \cup S$ **sinon** Soit $\Omega \in \mathcal{F}$ tel que $S \subseteq \Omega \subseteq C \cup S$ $\mathcal{T} \leftarrow$ l'arbre ayant comme unique étiquette Ω **si** $\Omega \neq C \cup S$ **alors** **pour chaque** $S' \in \Gamma_G(\Omega) \setminus \{S\}$ non inclus dans S et $C' \in C_G^*(\Omega)$ tels que $\Omega \not\subseteq C' \cup S'$ **faire** $\mathcal{T}' \leftarrow$ `extraction_aux`($G, S', C', \mathcal{F}, \Gamma$) créer une arête entre les racines de \mathcal{T}' et \mathcal{T} **rendre** \mathcal{T} **fin**

Algorithme 5.3 `extraction`

entrée : \mathcal{F} : une famille complète de blocs d'un hyper-graphe G connexe. Γ : l'ensemble des séparateurs qui bordent \mathcal{F} **sortie :** \mathcal{T} : un arbre des cliques d'une triangulation H de G dont les cliques maximales sont des blocs de \mathcal{F} **début****rendre** `extraction_aux`($G, \emptyset, V, \mathcal{F}, \Gamma$)**fin**

Propriété 5.8 *L'algorithme `extraction_aux` est correct.*

□ Notons \mathcal{F}_X l'ensemble des blocs de \mathcal{F} inclus dans X . Montrons la propriété par induction sur le nombre de blocs de $\mathcal{F}_{C \cup S}$.

Considérons Ω le bloc choisi par `extraction_aux`. Si Ω est égal à $C \cup S$, l'arbre rendu vérifie bien les conditions voulues.

Sinon, soit x dans $C \cup S$. Le sommet x appartient soit à Ω , soit à une composante connexe C' de $G \setminus \Omega$. Le voisinage S' de cette composante connexe est un séparateur minimal de G qui borde Ω et qui n'est pas inclus dans S . Comme $\mathcal{F}_{C' \cup S'}$ est inclus dans $\mathcal{F}_{C \cup S}$ et qu'il ne contient pas Ω , par induction, l'arbre rendu par l'appel récursif à `extraction_aux` sur S' et C' rend bien un arbre des cliques vérifiant les conditions requises.

Plus généralement, pour chaque S' bordant Ω qui n'est pas inclus dans S et chaque composante pleine C' dans $\mathcal{C}_G^*(S')$ telle que $C' \cup S'$ ne contienne pas Ω , $\mathcal{F}_{C' \cup S'}$ est inclus dans $\mathcal{F}_{C \cup S}$ et l'arbre produit par C' et S' vérifie bien les conditions voulues.

Au final, l'arbre \mathcal{T} est bien un arbre des cliques enraciné dont les étiquettes sont des blocs de $\mathcal{F}_{C \cup S}$. Sa racine Ω contient S et chaque sommet de $C \cup S$ apparaît dans une étiquette de \mathcal{T} . Il vérifie donc toutes les conditions requises. ■

Corollaire 5.9 *Si \mathcal{F} est une famille complète de blocs d'un hyper-graphe connexe, il existe une triangulation de G dont les cliques maximales sont des blocs de \mathcal{F} . L'algorithme `extraction` est calculé une.*

Ces deux algorithmes sont exactement ceux décrits dans la thèse de Todinca [Tod99] mais présentés dans un cadre plus général. Il donne la propriété suivante :

Propriété 5.10 *Si nous notons r le nombre de séparateurs minimaux d'un hyper-graphe G , b le nombre de couples (C_S, S) où S est un séparateur minimal de G et C_S une composante pleine dans $\mathcal{C}_G^*(S)$ et p le nombre de blocs, la complexité de l'algorithme `elimination` est $O(prn + bn)$ et celle de l'algorithme `extraction` est $O(prn \log n)$.*

Comme b est au plus égal à nr et r au plus à np , ces algorithmes fonctionnent en temps polynomial en fonction du nombre de blocs de la famille d'entrée.

Sommaire

6.1	Largeur arborescente	65
6.2	Largeur de branches	66
6.2.1	Hyper-graphes et triangulations	67
	Un algorithme de décision	69
6.2.2	Largeur de branches des cliques et hyper-cliques	71
6.2.3	Profil	75
6.2.4	Le théorème de décomposition serrée	80

Au chapitre 4, nous avons montré que la largeur de branches d'un hyper-graphe est égale à la plus petite largeur de branches d'une triangulation de G et qu'il en est de même pour la largeur arborescente.

Dans ce chapitre, nous montrons que nous pouvons nous restreindre aux triangulations serrées introduites au chapitre 5. De plus, nous montrons comment adapter les algorithmes de calcul de triangulations du chapitre 5 pour retrouver l'algorithme de décision de la largeur arborescente de Bouchitté et Todinca [BT01a] d'une part et introduire un algorithme analogue pour la largeur de branches d'autres part.

6.1 Largeur arborescente

Propriété 6.1 (Parra et Scheffler) *Une triangulation minimale est une triangulation serrée.*

□ Soient H une triangulation minimale d'un hyper-graphe G et S un séparateur minimal de H . Montrons que les ensembles $C_G(S)$ et $C_G^*(S)$ sont respectivement égaux à $C_H(S)$ et $C_H^*(S)$. Notons que cette condition permet aussi d'affirmer que S est un séparateur minimal de G .

Comme G est un sous-hyper-graphe de H , les composantes connexes de $H \setminus S$ sont des réunions de composantes connexes de $G \setminus S$. Pour chaque composante C_i de $C_G(S)$, considérons H_i l'hyper-graphe $H[C_i \cup N(C_i)]$. L'hyper-graphe H étant triangulé, l'hyper-graphe H_i l'est aussi. Soient H' la réunion des hyper-graphes H_i et de $H[S]$ et μ un cycle de longueur quatre ou plus de H' . Si μ est inclus dans l'un des H_i , comme ceux-ci sont triangulés, μ admet une corde. Dans le cas contraire, μ possède au moins deux sommets de S non adjacents sur le cycle et comme d'après la propriété 2.22 S induit

une clique de H , μ admet une corde. L'hyper-graphe H' est donc triangulé. De plus, les extrémités d'une hyper-arête e de G appartiennent soit à S , soit à l'un des $C_i \cup N(C_i)$. Par conséquent, e appartient à $H[S]$ ou à l'un des H_i . L'hyper-graphe H' est donc une triangulation de G . De plus, c'est un sous-hyper-graphe de H . La minimalité de H assure que H et H' sont égaux. Finalement, les ensembles $C_G(S)$ et $C_G^*(S)$ sont respectivement égaux, par construction, à $C_H(S)$ et $C_H^*(S)$.

Le séparateur minimal S étant quelconque, l'hyper-graphe H est bien une triangulation serrée de G . ■

La propriété 6.1 assure que nous pouvons nous restreindre aux triangulations serrées de G pour calculer la largeur arborescente de G . De plus, la largeur arborescente d'une triangulation H de G est la plus grande taille d'une clique maximale de H et donc, si H est serrée, $\text{tw}(H) + 1$ est la plus grande taille d'un bloc de G que H utilise. Nous en déduisons donc le théorème suivant dû à Bouchitté et Todinca :

Théorème 6.2 ([BT01a])

Si nous pouvons énumérer en temps polynomial toutes les cliques maximales potentielles d'une classe d'hyper-graphes, la largeur arborescente se calcule en temps polynomial.

□ Soit \mathcal{F} l'ensemble des cliques maximales potentielles d'un hyper-graphe G . Pour savoir si la largeur arborescente de G est au plus k , il suffit de considérer la sous-famille \mathcal{F}_k de \mathcal{F} constituée des cliques maximales potentielles de \mathcal{F} de taille au plus $k + 1$. Si \mathcal{F}_k est complète, la largeur arborescente de G est au plus k . ■

D'autre part, Bouchitté et Todinca [BT01b] montrent qu'il est possible d'énumérer les cliques maximales potentielles de G en temps polynomial en fonction du nombre de séparateurs minimaux de G . Comme il est possible d'énumérer efficacement les séparateurs minimaux, ils en déduisent le théorème suivant :

Théorème 6.3 ([BT01b])

Le calcul de la largeur arborescente est polynomial sur toute classe d'hyper-graphes ayant un nombre polynomial de séparateurs minimaux.

6.2 Largeur de branches

Dans cette partie, nous adaptons à la largeur de branches l'algorithme *elimination* en utilisant la même démarche que pour la largeur arborescente. Cependant, ceci s'avère plus difficile.

Dans un premier temps, nous montrons comment obtenir un algorithme qui, à partir d'une famille de blocs d'un hyper-graphe G et de leurs *largeurs*

de branches, calcule à la volée une majoration de la largeur de branches d'une triangulation serrée de G . Ensuite, nous étudions les conditions nécessaires pour que ce majorant soit la largeur de branches. Nous devons obtenir une famille de blocs de G suffisamment grande dont nous pouvons calculer les largeurs de branches ; de plus, la largeur de branches de G doit être égale à la plus petite largeur de branches d'une de ses triangulations serrées. Ceci nous amène à évaluer la largeur de branches des hyper-cliques puis à introduire la notion de profil qui nous permet de prouver que la seconde hypothèse est toujours vérifiée.

6.2.1 Hyper-graphes et triangulations

Définitions 6.4 (Largeur de branches d'un bloc)

Soit Ω un bloc d'un hyper-graphe G bordé par les séparateurs minimaux $(S_i)_{i \in I}$. L'hyper-graphe $G[\Omega]$ auquel nous ajoutons les hyper-arêtes $(S_i)_{i \in I}$ est l'hyper-graphe G_Ω du bloc Ω dans G .

La largeur de branches du bloc Ω est celle de G_Ω .

D'après le théorème 2.35, chaque clique maximale d'un hyper-graphe triangulé H est un bloc. Ceci nous permet de parler de la *largeur de branches d'une clique maximale* de H . La propriété suivante relie la largeur arborescente d'un hyper-graphe triangulé à celles de ses cliques maximales.

Propriété 6.5 *Soit H un hyper-graphe triangulé. Il existe une décomposition en branches de H dont la largeur est la plus grande largeur de branches d'une de ses cliques maximales.*

□ Soient \mathcal{T} un arbre des cliques de H , $(\Omega_i)_{i \in I}$ les cliques maximales de H et $(T_{i\tau_i})_{i \in I}$ des décompositions des hyper-graphes H_{Ω_i} de largeurs minimales.

Démontrons la propriété par induction sur le nombre de cliques maximales de H . Si H ne contient qu'une clique maximale Ω_1 , H est égal à Ω_1 et la décomposition en branches $T_{1\tau_1}$ convient.

Sinon, considérons une arête e de \mathcal{T} correspondant à un séparateur minimal S minimal pour l'inclusion. Cette arête induit deux composantes connexes de $\mathcal{T} \setminus \{e\}$ et donc une partition des cliques maximales de H en deux ensembles $(\Omega_j)_{j \in J}$ et $(\Omega_i)_{i \in I \setminus J}$. Soient H_1 et H_2 les hyper-graphes $H[\cup_{j \in J} \Omega_j]$ et $H[\cup_{i \in I \setminus J} \Omega_i]$ dans lesquels nous avons remplacé les arêtes de $H[S]$ par l'hyper-arête S . Les cliques maximales de H sont exactement celles de H_1 et de H_2 .

Le nombre de cliques maximales de H_1 et de H_2 est strictement plus petit que celui de H . Il existe donc deux décompositions en branches $T_{1\tau_1}$ de H_1 et $T_{2\tau_2}$ de H_2 dont les largeurs respectives sont les plus grandes largeurs de branches des blocs de H_1 et de H_2 . Dans chacune de ces deux décompositions, il existe une feuille correspondant à l'hyper-arête S . En retirant ces feuilles, nous obtenons deux branches $*T_1$ et $*T_2$. Construisons

une branche $*T_3$ de support S . Nous pouvons relier ces trois branches en une décomposition en branches T_τ de H . Montrons que T_τ vérifie les conditions requises.

Soit Ω une clique maximale de H_1 . Comme S est minimal pour l'inclusion, le seul séparateur minimal bordant Ω dans H pouvant ne pas border Ω dans H_1 est S . Mais dans ce cas, comme S est inclus dans la clique Ω , la largeur de branches de Ω est la même dans H et dans H_1 . La plus grande largeur de branches d'une clique maximale de H est la plus grande largeur de branches d'une clique maximale de H_1 et de H_2 . Elle vaut donc $\max(\text{bw}(T_{1\tau_1}), \text{bw}(T_{2\tau_2}))$.

Les arêtes ajoutées à $*T_1$ et $*T_2$ correspondent à des frontières incluses dans S dont la largeur est inférieure ou égale à la plus grande largeur de branches d'une clique maximale de H . Soit f une arête de T correspondant à une arête de $*T_1$. La frontière F correspondant à f dans T_τ est égale à celle F_1 de f dans $T_{1\tau_1}$. En effet, par construction, F_1 est incluse dans F et comme les seuls sommets incidents à la fois à une hyper-arête de H_1 et à une hyper-arête de H_2 sont ceux de S , F est inclus dans F_1 . La largeur de branches de T_τ vaut $\max(\text{bw}(T_{1\tau_1}), \text{bw}(T_{2\tau_2}))$, ce qui termine la démonstration. ■

Pour résumer la démonstration de la propriété 6.5, nous pouvons dire qu'à partir des décompositions en branches des hyper-graphes H_{Ω_i} , nous construisons une décomposition en branches de H en les « recollant » selon un arbre des cliques. Cette propriété ne nous donne qu'un majorant de la largeur de branches de H comme l'énonce la propriété suivante :

Propriété 6.6 *Il existe un graphe triangulé H tel que toute décomposition en branches de H de largeur minimale induise un sur-graphe strict de H .*

□ Soit H obtenu à partir d'une clique de taille n à laquelle nous ajoutons deux sommets u et v reliés à tous les sommets de la clique. Nous pouvons aussi voir H comme K_{n+2} auquel nous avons retiré une arête. La clique est un u, v -séparateur minimal.

D'après le corollaire 6.10, la largeur de branches de H est au plus égale à $\lceil 2(n+2)/3 \rceil$; or si une décomposition T de H n'induit pas de sur-graphe strict de H , une arête de T correspond à la clique K_n de départ. La largeur de T est donc au moins n . ■

Cette propriété distingue la largeur de branches de la largeur arborescente. En effet, parmi les décompositions arborescentes de largeur minimale d'un hyper-graphe G , il y en a toujours une qui induit une triangulation minimale de G et donc si G est triangulé, l'arbre de cliques de G fournit une décomposition arborescente optimale.

Un algorithme de décision

Nous pouvons maintenant appliquer les algorithmes de calcul de triangulations du chapitre 5 à la décision de la largeur de branches. En effet, considérons une famille \mathcal{F} de blocs de G dont nous connaissons les largeurs de branches. Définissons la sous-famille \mathcal{F}_k des blocs de \mathcal{F} dont la largeur de branches est inférieure ou égale à k . Si la famille \mathcal{F}_k contient une famille complète, il existe une triangulation serrée H de G dont les cliques maximales sont des blocs de \mathcal{F}_k . D'après la propriété 4.18, la largeur de branches de G est inférieure ou égale à celle de H et d'après la propriété 6.5, celle de H est inférieure ou égale à la plus grande largeur de branches d'une de ses cliques maximales. Or celle des blocs de \mathcal{F}_k est inférieure ou égale à k . Nous avons donc :

$$\text{bw}(G) \leq \text{bw}(H) \leq k.$$

Algorithme 6.1 `bw_elimination`

entrée :

k : un entier

\mathcal{F} : une famille de blocs d'un hyper-graphe G
dont nous connaissons les largeurs de branches.

$\Gamma_{\mathcal{F}}$: l'ensemble des séparateurs qui bordent \mathcal{F}

sortie :

\mathcal{F}_k : la plus grande famille complète de blocs
de largeur au plus k incluse dans \mathcal{F}

début

tant que $\exists \Omega \in \mathcal{F}$ tel que $\text{bw}(\Omega) > k$ **faire**

$\mathcal{F} \leftarrow \mathcal{F} \setminus \{\Omega\}$

mettre à jour $\Gamma_{\mathcal{F}} = \bigcup_{\Omega \in \mathcal{F}} \Gamma_{\Omega}$

$\mathcal{F} \leftarrow \text{elimination}(\mathcal{F}, \Gamma_{\mathcal{F}})$

rendre \mathcal{F}

fin

De cette façon, nous obtenons l'algorithme `bw_elimination`. Si celui-ci renvoie une famille complète non vide, il existe une décomposition en branches de G de largeur au plus k . Ceci assure que la largeur de branches de G est inférieure ou égale à k . Malheureusement, s'il renvoie une famille vide, nous ne pouvons pas conclure que la largeur de branches de G est strictement supérieure à k car les inégalités précédentes peuvent être strictes.

Comme, d'après la propriété 4.26, la largeur de branches d'un hyper-graphe G est égale à la plus petite largeur de branches d'une de ses triangulations, la première inégalité peut être une égalité. En fait, la triangulation H et la décomposition en branches de H construites par cette propriété per-

mettent d'affirmer que les deux inégalités peuvent être simultanément des égalités :

Propriété 6.7 *Soit H une triangulation d'un hyper-graphe G induite par une décomposition en branches de G de largeur minimale.*

La largeur de branches de G est égale à celle de H et à la plus grande largeur de branches d'une des cliques maximales de H .

□ L'hyper-graphe H est induit par une décomposition en branches T_τ de largeur minimale de G .

Reprenons la construction de la décomposition en branches de H à partir de celle de T_τ détaillée à la propriété 4.26. Pour chaque arête xy de H n'apparaissant pas dans G , nous avons montré qu'il existe une arête e de T_τ dont la frontière contient simultanément x et y . En ajoutant un nœud v au milieu de e , nous pouvons ajouter une feuille reliée à v dans l'arbre T . L'étiquette de cette feuille contient l'arête xy .

De cette manière, comme les deux sommets x et y appartiennent déjà à la frontière de e , nous n'augmentons pas la largeur de branches. De plus, comme x et y appartiennent déjà à une même frontière, les sous-arbres T_x et T_y induits pas les frontières contenant respectivement x et y ont une intersection non vide. La triangulation induite par notre nouvel arbre ne contient donc pas d'arête supplémentaire par rapport à H . En répétant ce procédé, nous obtenons une décomposition en branches T'_τ de H dont la largeur de branches est celle de G et telle que la triangulation qu'elle induit soit égale à H .

Soit Ω une clique maximale de H . Montrons que la largeur de branches de H_Ω est inférieure ou égale à celle de H . La triangulation H étant induite par la décomposition en branches T'_τ , pour chaque séparateur minimal S bordant Ω , il existe une arête e_S de T' telle que la frontière de e' soit égale à S . Ajoutons un nœud u_S au milieu de e_S et relierons u_S à une nouvelle feuille dont l'étiquette est S . De cette façon, nous obtenons une décomposition en branches de l'hyper-graphe H' égal à l'hyper-graphe H auquel nous avons ajouté les hyper-arêtes correspondant aux séparateurs minimaux bordant Ω . La largeur de la décomposition obtenue est exactement $\text{bw}(H)$. Or H_Ω est un sous-hyper-graphe de H' . Ceci prouve que la largeur de branches de la clique Ω est inférieure ou égale à celle de H . En utilisant la propriété 6.5, nous obtenons que la largeur de branches de H est égale à la plus grande largeur de branches d'une clique maximale de H . ■

À partir des résultats précédent, nous pouvons conclure :

Propriété 6.8 *S'il existe une décomposition en branches de G de largeur minimale qui induit une triangulation serrée de G et si la famille \mathcal{F} contient les blocs d'une telle décomposition, la propriété 6.7 nous assure que bw_eli-*

mination rend une famille complète vide si et seulement si la la largeur de branches de G est strictement supérieure à k .

Cette propriété s'appuie sur deux hypothèses : l'existence d'une triangulation serrée et la possibilité de calculer la largeur de branches des blocs de la famille \mathcal{F} . Dans la suite, nous nous intéressons à la largeur de branches de cliques puis nous montrons qu'il existe toujours une décomposition en branches de largeur minimale induisant une triangulation serrée de G . De cette façon, si nous savons construire une famille de blocs adaptée, nous pouvons utiliser `bw_elimination` pour décider la largeur de branches.

6.2.2 Largeur de branches des cliques et hyper-cliques

Si le calcul de la largeur arborescente pour les hyper-graphes complets est trivial ($\text{tw}(G)$ est égal à $|V(G)| - 1$), nous montrons ici que la situation est complètement différente pour la largeur de branches. En effet, le calcul de la largeur de branches pour les hyper-cliques est NP-dur. Néanmoins, il reste polynomial pour des hyper-cliques particulières, propriété qui nous sert fortement pas la suite.

Propriété 6.9 ([RS91]) *La largeur de branches $\text{bw}(K_n)$ d'une clique (le graphe complet à n sommets) de taille supérieure ou égale à trois est égal à $\lceil 2n/3 \rceil$ pour n au moins égal à trois.*

□ Soit T_τ une décomposition en branches de K_n de largeur minimale. Cette décomposition induit une décomposition arborescente T_{θ_τ} de K_n . Par conséquent, il existe un nœud v de T dont l'étiquette $\theta_\tau(v)$ contient tous les sommets de K_n . Soient S_1, S_2 et S_3 les frontières de T_τ correspondant aux arêtes incidentes à v .

Comme les étiquettes des feuilles de T_τ sont des arêtes, elles sont de taille deux. Le nœud v est donc un nœud interne de T . D'après le lemme 4.19, chaque sommet de K_n apparaît dans au moins deux frontières bordant v , d'où $2n \leq |S_1| + |S_2| + |S_3| \leq 3 \text{bw}(G)$. La largeur de branches de K_n est donc supérieure ou égale à $\lceil 2n/3 \rceil$.

Pour finir la démonstration, il suffit d'exhiber une décomposition de largeur $\lceil 2n/3 \rceil$. Pour cela, numérotions les sommets de K_n de 1 à n et posons $A_1 = \{x_1, \dots, x_{\lfloor n/3 \rfloor}\}$, $A_2 = \{x_{\lfloor n/3 \rfloor + 1}, \dots, x_{\lfloor 2n/3 \rfloor}\}$ et $A_3 = \{x_{\lfloor 2n/3 \rfloor + 1}, \dots, x_n\}$. La taille des ensembles $A_i \cup A_j$ est au plus $\lceil 2n/3 \rceil$.

Soient B_i l'ensemble $A_i \times A_i$ des arêtes entre les sommets de A_i , $B_{i,j}$ l'ensemble $A_i \times A_j$ ($i \neq j$), $*T_i$ une branche quelconque sur B_i et $*T_{i,j}$ une branche quelconque sur $B_{i,j}$. La décomposition donnée par la figure 6.1 est de largeur $\lceil 2n/3 \rceil$. ■

Un corollaire immédiat de cette propriété est le suivant :

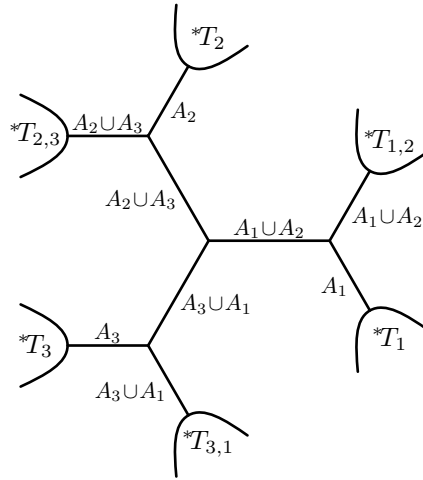


FIG. 6.1 – Décomposition en branches d'une clique

Corollaire 6.10 *La largeur de branches d'un graphe est au plus $\lceil 2n/3 \rceil$.*

□ En effet, tout graphe G à n sommets est un sous-graphe de K_n et d'après la propriété 4.18, $\text{bw}(G) \leq \text{bw}(K_n) = \lceil 2n/3 \rceil$. ■

La propriété 6.9 et le corollaire 6.10 montrent que la largeur de branches d'un graphe à n sommets est toujours strictement inférieure à n . En revanche, si nous considérons des hyper-graphes et en particulier des hyper-cliques, la situation est différente. Par exemple, si G est une clique à laquelle nous ajoutons l'hyper-arête V , quelle que soit la décomposition en branches, la taille de la frontière de l'hyper-arête V est toujours n . Cependant, cet exemple est extrême. Nous pourrions imaginer que si la taille de la plus grosse hyper-arête reste petite, la largeur de branches reste inférieure à n . Cette intuition est fautive comme le montre le contre-exemple développé dans la propriété 6.11.

Propriété 6.11 *La largeur de branches de l'hyper-clique $G = (V, E)$ où E est l'ensemble $\{\{x, y, z\} \in V^3\}$ est $|V|$.*

□ Supposons par l'absurde que T_τ soit une décomposition en branches de G dont la largeur est strictement inférieure à $|V|$. Soit T_{θ_τ} la décomposition arborescente induite par T_τ .

L'hyper-graphe G étant une clique, soient u un nœud de T dont l'étiquette $\theta_\tau(u)$ est V , S_1 , S_2 et S_3 les trois frontières de T_τ correspondant aux arêtes de T incidentes à u et E_1 , E_2 et E_3 les supports de branches associés. Comme $S_1 \cup S_2 \cup S_3$ est égal à V , tout sommet x de V appartient à au moins une frontière de S_i . De plus, u étant un nœud interne de T , d'après le lemme 4.19, x appartient à deux frontières S_i et S_j . Nous en déduisons donc que

si le sommet x de V n'appartient pas à la frontière S_i , il n'est inclus dans aucune arête de E_i . En revanche, il appartient aux deux autres frontières S_j .

Nous avons supposé que la largeur de branches de T_τ était strictement inférieure à $|V|$. Il existe donc x_1, x_2 et x_3 trois sommets distincts tels que x_i n'appartienne à aucune arête de E_i . Mais alors l'hyper-arête $\{x_1, x_2, x_3\}$ n'apparaît dans aucun des ensembles E_i ce qui est absurde. La largeur de branches de T est donc forcément $|V|$. ■

La problème de la largeur de branches des hyper- cliques est donc plus compliqué que celui des cliques. En fait, Kloks et col. [KKM99] démontrent le théorème suivant :

Théorème 6.12 ([KKM99])

Soit $G = (I, C, E)$ un split-graph¹⁰. Décider si la largeur de branches de G vaut $\lceil 2|C|/3 \rceil$ est un problème NP-complet.

Si $G = (I, C, E)$ est un split-graphe et si nous remplaçons chaque sommet x de I par une hyper-arête dont les sommets sont ceux du voisinage de x , nous obtenons une hyper-clique K . La largeur de branches de K vaut $\lceil 2|K|/3 \rceil$ si et seulement si $\text{bw}(G)$ est égal à $\lceil 2|C|/3 \rceil$. Le théorème 6.12 prouve donc que le problème de décision associé à la largeur de branches est NP-complet pour les hyper- cliques.

Nous ne démontrons pas ce théorème ici mais nous allons donner une idée du problème sous-jacent. Considérons une clique de taille $3k$ et une partition de ses sommets en trois ensembles A_1, A_2 et A_3 tous de taille k . Nous pouvons construire une décomposition comme celle de la figure 6.1 de telle sorte que les frontières S_1, S_2 et S_3 qui bordent le nœud central soient exactement $A_1 \cup A_2, A_2 \cup A_3$ et $A_3 \cup A_1$. Si sur cette clique, nous greffons des sommets dont le voisinage est inclus dans l'une des frontières S_i , alors nous pouvons placer les nouvelles arêtes dans la branche contenant les arêtes $B_i \cup B_{i,i+1}$ sans augmenter la largeur de la décomposition. Dans ce cas, il est possible de partitionner l'ensemble des sommets greffés en trois suivant la frontière à laquelle ils se rattachent. Si pour toute décomposition de la clique, il existe un sommet « à cheval » sur plusieurs frontières, la largeur de branches doit être strictement supérieure à $2k$.

Si une hyper-clique admet un nombre borné d'hyper-arête de taille trois ou plus, nous montrons comment, en énumérant toutes les partitions en trois de ces « grosses » hyper-arêtes, nous pouvons calculer la largeur de branches de l'hyper-clique.

¹⁰Un split-graph est un graphe $G = (I, C, E)$ tel que $G[C]$ soit une clique et tel qu'aucune arête ne relie deux sommets de I . Les split-graphs sont des graphes triangulés.

Théorème 6.13

Soit G une hyper-clique ayant k hyper-arêtes f_1, \dots, f_k de taille trois ou plus. La largeur de branches de G peut se calculer en temps $O(3^k)$.

□ Soient T_τ une décomposition en branches dont la largeur est $\text{bw}(G)$, T_{θ_τ} la décomposition arborescente induite, u un nœud de T dont l'étiquette $\theta_\tau(u)$ est V et $*T_1, *T_2$ et $*T_3$ les trois branches de $T \setminus \{u\}$.

Les hyper-arêtes f_i se répartissent en trois ensembles F_1, F_2 et F_3 éventuellement vides correspondant aux branches dans laquelle elles se trouvent. La largeur de branches de G est la même que celle de l'hyper-graphe G dans lequel nous avons contracté les hyper-arêtes de F_i en une grosse hyper-arête e_i . Donc si nous savons calculer la largeur de branches de G quand il possède au plus trois « vraies » hyper-arêtes en temps $O(1)$, il suffit de tester les 3^k répartitions possibles des hyper-arêtes f_i en trois ensembles F_j .

Montrons que la largeur de branches de G peut se calculer en temps $O(1)$ quand il possède au plus trois hyper-arêtes de taille trois ou plus.

Notons e_1, e_2 et e_3 trois plus grosses hyper-arêtes de G . Par hypothèse, toute hyper-arête de taille trois ou plus est l'une des hyper-arêtes e_i . Il n'existe une décomposition de largeur k que si nous pouvons trouver une bonne partition $A \sqcup B \sqcup C$ de V . Soient α, β et γ les tailles respectives de A, B et C . Une telle décomposition existe si et seulement si α, β et γ vérifient le système suivant :

- $\alpha > 0; \beta > 0; \gamma > 0;$
- $\alpha + \beta + \gamma = n;$

Ces conditions assurent que A, B et C forment une partition de V .

- $\alpha + \beta \leq k; \beta + \gamma \leq k; \gamma + \alpha \leq k;$

Ces conditions assurent que la largeur de la décomposition est inférieure à k .

- $\alpha + \beta \geq |e_1|; \beta + \gamma \geq |e_2|; \gamma + \alpha \geq |e_3|;$

Ces conditions assurent que e_1, e_2 et e_3 peuvent être incluses respectivement dans $A \cup B, B \cup C$ et $C \cup A$.

- $\beta \geq |e_1 \cap e_2|; \gamma \geq |e_2 \cap e_3|; \alpha \geq |e_3 \cap e_1|.$

La dernière condition assure que $e_1 \cap e_2, e_2 \cap e_3$ et $e_3 \cap e_1$ peuvent être inclus respectivement dans B, C et A .

Ce système est analogue à celui donné dans la démonstration du lemme 9. de l'article de Kloks et col. [KKM99] qui prouve que le calcul de la largeur de branches des graphes d'intervalles est polynomial.

Le système précédent est un programme linéaire paramétrique qui se résout paramétriquement en temps polynomial [Fea88]¹¹. Une fois la solution obtenue, calculer la largeur de branches d'une hyper-clique ayant au

¹¹Le programme PIP de Paul Feautrier résout ce type de systèmes. Il se trouve à l'adresse <http://www.prism.uvsq.fr/~cedb/bastools/piplib.html>.

plus trois hyper-arêtes de taille supérieure ou égale à trois se fait en temps constant. ■

6.2.3 Profil

Les triangulations induites par une décomposition en branches de largeur minimale d'un hyper-graphe G peuvent ne jamais être minimales. Cependant, nous allons montrer dans cette partie qu'il en existe toujours une qui induit une triangulation serrée. Pour cela, nous allons nous intéresser plus avant aux frontières que définit une décomposition en branches et montrer qu'il est toujours possible de transformer une décomposition en branches en une décomposition serrée sans augmenter sa largeur.

L'outil fondamental que nous introduisons ici pour affiner une décomposition est la notion de *profil* qui prend en compte les tailles de presque toutes les frontières de la décomposition. Nous montrons que toutes les décompositions de profil minimal (en particulier certaines de largeur de branches minimale) induisent des triangulations serrées.

Définitions 6.14 (Paquet)

Soient S une frontière d'un hyper-graphe $G = (V, E)$ et A inclus dans E dont la frontière est égale à S . Un ensemble P d'hyper-arêtes de E est un paquet de S si l'une des deux conditions suivantes est vérifiée :

- P est un singleton contenant une hyper-arête dont toutes les extrémités sont dans S ;
- il existe une composante connexe C de $G \setminus S$ telle que P soit l'ensemble des hyper-arêtes de G incidentes à au moins un sommet de C .

Un paquet de A est un paquet de S inclus dans A . Un paquet P induit par une composante connexe capture un ensemble de sommets S si tous les sommets de S sont incidents à au moins une arête de P .

Le lemme 6.15 justifie l'introduction de la notion de paquet en montrant comment les frontières s'articulent autour des paquets.

Lemme 6.15 Soient S une frontière d'un hyper-graphe G , A inclus dans $E(G)$ dont la frontière est égale à S et P un paquet de S . Soit P est inclus dans A , soit P et A sont disjoints.

□ Si P est un singleton, l'affirmation est triviale. Sinon, il existe une composante connexe C de $G \setminus S$ telle que P soit l'ensemble des hyper-arêtes incidentes à au moins un sommet de C .

Supposons que e et f soient deux hyper-arêtes de P avec e dans A et f à l'extérieur de A . Par définition de P , il existe u et v dans C incidents respectivement à e et f . Puisque C est une composante connexe de $G \setminus S$, il existe une chaîne $\mu_{u \rightarrow v}$ de u à v dans $G[C]$. La chaîne $e.\mu_{u \rightarrow v}.f$ égale à $(x_0, e_0, \dots, e_{l-1}, x_l)$ commence dans A et finit dans $E \setminus A$. Par conséquent, il

existe une première hyper-arête e_i le long de la chaîne qui appartient à $E \setminus A$. Comme l'hyper-arête e_{i-1} n'appartient pas à $E \setminus A$, le sommet x_{i-1} est un sommet de la frontière S , ce qui est absurde. ■

La notion de paquet permet une relecture de la définition de parallélisme sur les ensembles de sommets :

Définition 2.17 bis (Parallélisme de frontières)

Soient S et T deux frontières d'un hyper-graphe G . L'ensemble S est parallèle à T si S est inclus dans T ou si S est capturé par un paquet de T .

Définition 6.16 (Frontière essentielle)

Une frontière est essentielle parmi un ensemble \mathcal{F} de frontières si aucune autre frontière de \mathcal{F} ne la contient.

Définition 6.17 (Profil)

Soit \mathcal{C} une matriochka complète d'un hyper-graphe G . Le profil de \mathcal{C} noté $\text{profil}(\mathcal{C})$ est le n -uplet (x_n, \dots, x_1) où x_i est le nombre de frontières essentielles de \mathcal{C} de cardinal i . Nous ordonnons les décompositions complètes sur G par l'ordre lexicographique sur leur profil, c'est-à-dire $\mathcal{C}' < \mathcal{C}$ si

$$\exists i \in [1..n], \forall j \in [i + 1..n], (\text{profil}(\mathcal{C}')_j = \text{profil}(\mathcal{C})_j) \wedge (\text{profil}(\mathcal{C}')_i < \text{profil}(\mathcal{C})_i).$$

Le profil d'un graphe G est le plus petit profil d'une décomposition en branches de G .

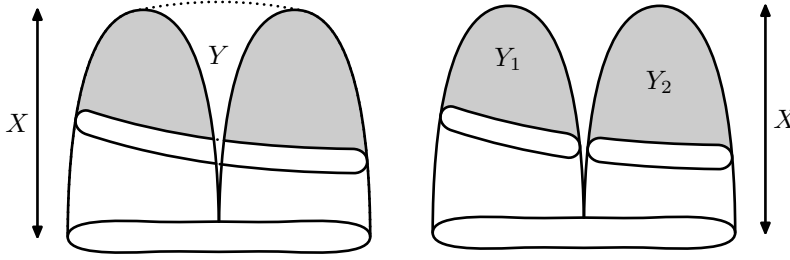
Si le profil de \mathcal{C}' est inférieur à celui de \mathcal{C} , alors la largeur de branches de \mathcal{C}' est inférieure à celle de \mathcal{C} .

Le problème de décision associé à la largeur de branches étant NP-complet, choisir une « bonne » frontière pour construire une décomposition en branches est difficile. Cependant, une fois une décomposition choisie, il semble possible de faire des optimisations locales pour l'améliorer comme le suggère la figure 6.2. Cependant, pour faire une modification sur un élément Y d'une matriochka \mathcal{C} , il faut modifier tous les éléments de \mathcal{C} inclus dans Y . Il n'est donc pas évident qu'il soit possible d'effectuer toutes ces transformations sans augmentation du profil. Nous montrons aux théorèmes 6.20 et 6.21 que cette intuition est correcte.

Définition 6.18 (Paquet morcelé)

Soit X un élément d'une matriochka \mathcal{C} . Un paquet P de X est morcelé dans \mathcal{C} s'il n'appartient pas à \mathcal{C} . Un élément de \mathcal{C} est morcelé si un de ses paquets l'est.

Le lemme suivant permet de comparer les profils de deux décompositions en branches.



Si X et Y appartiennent à une matryochka \mathcal{C} et que la frontière de Y croise celle de X , le profil de \mathcal{C} n'est pas minimal.
 Si nous répartissons Y en Y_1, Y_2, \dots selon les paquets de X , le profil diminue.

FIG. 6.2 – Croisement et profil

Lemme 6.19 Soient T_1 et T_2 deux décompositions en branches d'un hyper-graphe G . Si chaque frontière de T_1 est incluse dans une frontière de T_2 , alors

$$\text{profil}(T_1) \leq \text{profil}(T_2).$$

De plus, s'il existe une frontière de T_2 qui n'est incluse dans aucune frontière de T_1 , alors

$$\text{profil}(T_1) < \text{profil}(T_2).$$

Théorème et définitions 6.20

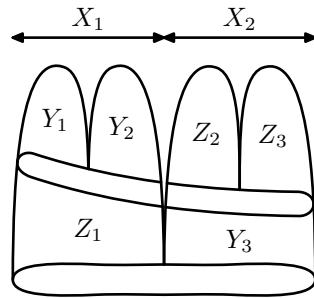
Soient T_τ une décomposition en branches d'un hyper-graphe G et $*T_X$ une branche de T_τ de support X . Il existe une décomposition en branches T'_τ , telle que :

- (i) T'_τ est obtenue en remplaçant $*T_X$ par une branche $*T'_X$ de même support ;
- (ii) X n'est pas morcelé dans T'_τ ;
- (iii) chaque frontière de T'_τ est incluse dans une frontière de T_τ ;
- (iv) $\text{profil}(T'_\tau) \leq \text{profil}(T_\tau)$.

La décomposition T'_τ est la décomposition T_τ peignée le long de X . Si e est une arête de T correspondant à une séparation $\{X, Y\}$, la décomposition obtenue en peignant T_τ le long de X et de Y est peignée le long de e .

□ Nous allons montrer le résultat par induction sur la taille de la branche $*T_X$. Si $|*T_X| = 1$, alors X n'est pas morcelé et T_τ vérifie les conditions requises.

Sinon, soient $*T_Y$ et $*T_Z$ les deux sous-branches de X . Par hypothèse d'induction, nous pouvons supposer que ni Y ni Z ne sont morcelés car sinon nous pourrions remplacer $*T_Y$ et $*T_Z$ de sorte que ni Y ni Z ne soient morcelés et cela sans augmenter le profil.



Considérer les branches correspondant aux paquets de Y et de Z et les réarranger selon les paquets de X .

Ainsi X n'est plus morcelé et le profil n'augmente pas.

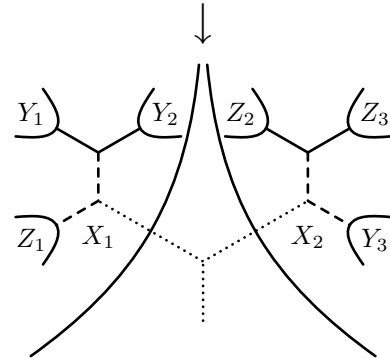
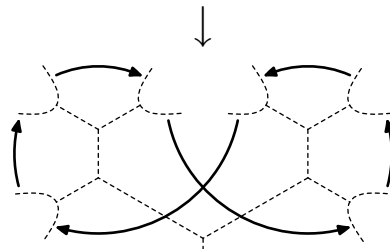
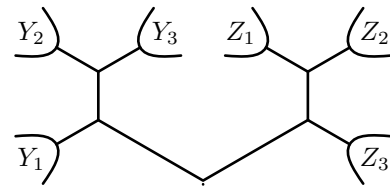
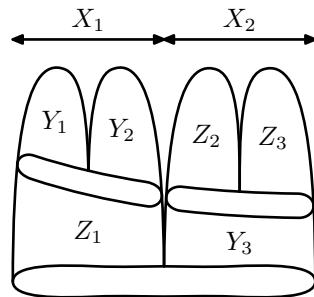


FIG. 6.3 – Fonctionnement du théorème 6.20

Soient $(X_k)_{k \in K}$ les paquets de X , $(Y_i)_{i \in I}$ ceux de Y et $(Z_j)_{j \in J}$ ceux de Z . Comme Y et Z sont strictement inclus dans X , les paquets de Y et de Z sont inclus dans des paquets de X . De plus, un paquet de Y ou de Z ne peut être inclus que dans un seul paquet de X . Nous pouvons donc partitionner I (resp. J) en $(I_k)_{k \in K}$ (resp. $(J_k)_{k \in K}$) de telle sorte que les paquets $(Y_i)_{i \in I_k}$ (resp. $(Z_j)_{j \in J_k}$) soient exactement les paquets de Y (resp. Z) inclus dans X_k . Construisons $*T'_X$ de la façon suivante :

- Pour chaque paquet X_k , définissons une branche $*T_{X_k}^Y$ (resp. $*T_{X_k}^Z$) à partir des sous-branches $(*T_{Y_i})_{i \in I_k}$ (resp. $(*T_{Z_j})_{j \in J_k}$);
- Soit $*T_{X_k}$ la branche dont $*T_{X_k}^Y$ et $*T_{X_k}^Z$ sont les deux sous-branches ;
- Finalement soit $*T'_X$ une branche quelconque obtenue à partir des branches $(*T_{X_k})_{k \in K}$

Nous affirmons que la décomposition en branches T'_τ , obtenue en remplaçant la branche $*T_X$ par $*T'_X$ satisfait les conditions requises. Par construction, T'_τ satisfait les conditions i et ii. Les arêtes de T' qui ne sont pas des arêtes de T sont :

- les arêtes utilisées pour construire $*T'_X$ à partir des branches $(*T_{X_k})_{k \in K}$. Or toutes ces arêtes correspondent aux frontières de réunions de paquets de X . Elles sont donc incluses dans $\partial(X)$;
- les arêtes utilisées pour construire les branches $*T_{X_k}^Y$ (resp. $*T_{X_k}^Z$). Or toutes ces arêtes correspondent aux frontières de réunions de paquets de Y (resp. Z) ; elles sont donc incluses dans $\partial(Y)$ (resp. $\partial(Z)$).

Donc toutes les frontières de T'_τ , sont incluses dans des frontières de T_τ ce qui montre que la condition iii est vérifiée et, d'après le lemme 6.19, la condition iv aussi. ■

Théorème 6.21

Soient T_τ une décomposition en branches d'un hyper-graphe G et e une arête de T correspondant à la frontière S_e . Si une frontière S de T_τ croise S_e , la décomposition T_τ peignée le long de e a un profil strictement plus petit que celui de T_τ .

□ Supposons qu'une frontière S de T_τ croise la frontière S_e correspondant à l'arête e de T . Elle n'est ni incluse dans S_e , ni capturée par un paquet de S_e . Soient $\{X, Y\}$ la séparation de T_τ correspondant à l'arête e et T'_τ , la décomposition T_τ peignée le long de e . D'après le théorème 6.20, toutes les frontières de T'_τ , sont incluses dans des frontières de T_τ et ni X , ni Y ne sont morcelés dans T'_τ .

Supposons qu'une frontière U de T'_τ , contienne S . Comme S n'est pas incluse dans S_e , U intersecte une composante connexe C de $G \setminus S_e$. Or par construction, la composante connexe C correspond à un paquet de S_e et donc à une branche $*T_C$ de T'_τ . La frontière U est donc une frontière de $*T_C$, ce qui est absurde car cela impliquerait que S soit capturée par C . Aucune

frontière de T'_τ , ne contient donc S . Nous pouvons appliquer le lemme 6.19 ce qui achève la démonstration. ■

Nous en déduisons comme corollaire immédiat :

Théorème 6.22 (Théorème de décomposition parallèle)

Toute décomposition en branches de profil minimal est parallèle.

Au chapitre 2, nous avons choisi une notion de parallélisme pour les frontières mais nous aurions aussi pu choisir la définition plus forte suivante :

Définition 6.23 (Parallélisme fort)

Soient S et T deux ensembles de sommets d'un hyper-graphe $G = (V, E)$. Nous disons que S est fortement parallèle à T s'il existe une composante connexe C de $G \setminus T$ avec S inclus dans $C \cup N(C)$. Si S n'est pas fortement parallèle à T , nous disons que S croise faiblement T .

Tout comme nous avons défini des décompositions en branches parallèles, nous pouvons définir des décompositions en branches fortement parallèles. Malheureusement, il n'existe pas d'analogue du théorème 6.22. La contre-exemple que nous avons introduit à la propriété 6.11 n'admet aucune décomposition fortement parallèle comme le justifie la propriété suivante :

Propriété 6.24 *Une hyper-clique $G = (V, E)$ où $|V| \geq 7$ et E est l'ensemble $\{\{x, y, z\} \in V^3\}$ n'admet pas de décomposition en branches dont les frontières sont toutes fortement parallèles deux à deux.*

□ L'hyper-clique G est celle de la propriété 6.11. Sa largeur de branches est $|V|$. Considérons maintenant une décomposition en branches T_τ de G . L'ensemble V est une frontière de T_τ . De plus, il existe dans T_τ une *cerise*, c'est-à-dire un nœud relié à deux feuilles. Soit A la réunion des deux hyper-arêtes associées à ces deux feuilles. La frontière de A est au plus de taille 6 et donc strictement plus petite que $|V|$; elle n'est capturée par aucun paquet de V . Elle n'est pas fortement parallèle à V . ■

Il est possible d'adapter ce contre-exemple pour montrer qu'il n'existe pas de théorème de décomposition fortement parallèle même pour les graphes.

6.2.4 Le théorème de décomposition serrée

Nous déduisons des théorèmes 6.20 et 6.21 quelques théorèmes dont nous avons besoin par la suite puis nous énonçons le théorème de décomposition serrée qui justifie que nous pouvons bien utiliser l'algorithme `bw_elimination` pour décider la largeur de branches d'un hyper-graphe.

Propriété 6.25 *Soient T_τ une décomposition en branches d'un hyper-graphe G et X le support d'une branche de T_τ .*

Si il existe une réunion Y de paquets de $\partial(X)$ telle que

(i) Y chevauche X ;

(ii) la réunion des frontières des paquets de X inclus dans Y et celle des paquets de X inclus dans $E \setminus Y$ soit strictement incluse dans $\partial(X)$;

alors il existe une décomposition en branches T'_τ de profil strictement plus petit que celui de T_τ .

□ D'après le théorème 6.20, nous pouvons supposer sans augmenter le profil que ni X ni $E \setminus X$ ne sont morcelés.

Soient $(Y_i)_{i \in I}$ les paquets de la frontière S de X et $({}^*T_{Y_i})_{i \in I}$ les branches de T_τ correspondantes. Supposons qu'il existe un sous-ensemble strict J de I tel que la réunion Y des paquets $(Y_j)_{j \in J}$ vérifie les conditions (i) et (ii). Construisons une branche *T_Y à partir des sous-branches $({}^*T_{Y_j})_{j \in J}$ et une seconde *T_Z à partir des sous-branches $({}^*T_{Y_i})_{i \in I \setminus J}$ puis relierons ces deux branches pour former une décomposition en branches T'_τ , comme l'illustre la figure 6.4.

Les arêtes de T' n'apparaissant pas dans T correspondent à des réunions de paquets de $\partial(X)$. Leurs frontières sont donc incluses dans $\partial(X)$. Chaque frontière de T'_τ est donc incluse dans une frontière de T_τ . Comme la réunion des frontières des paquets $(Y_j)_{j \in J}$ est strictement incluse dans $\partial(X)$, il existe des sommets de $\partial(X)$ qui ne sont pas capturés par Y . Aucune frontière de la branche *T_Y ne peut donc contenir $\partial(X)$. De même, aucune frontière de la branche *T_Z ne peut contenir $\partial(X)$. D'après le lemme 6.19,

$$\text{profil}(T'_\tau) < \text{profil}(T_\tau). \quad \blacksquare$$

Corollaire 6.26 Soient T_τ une décomposition en branches d'un hyper-graphe G et S une frontière de T_τ .

Pour x dans S , on pose X_S^x la réunion de l'ensemble des paquets de S ayant x dans leur frontière. S'il existe x tel que $\partial(X_S^x)$ soit strictement incluse dans S , alors il existe une décomposition en branches de G dont le profil est strictement plus petit que celui de T_τ .

□ Soient X dans \mathcal{C}_{T_τ} dont la frontière est S et x dans S tel que $\partial(X_S^x)$ soit strictement incluse dans S .

Puisque x appartient à S , il existe un paquet de X et un paquet de Y dont la frontière contient x . De même, il existe un paquet de X et un paquet de Y dont la frontière ne contient pas x . Ceci implique que X_S^x chevauche X . D'après le théorème 6.21, il existe une décomposition en branches de profil strictement plus petit que celui de T_τ . ■

Théorème 6.27

Soit T_τ une décomposition en branches parallèle de G . La triangulation H_{T_τ} de G induite par T_τ est serrée.

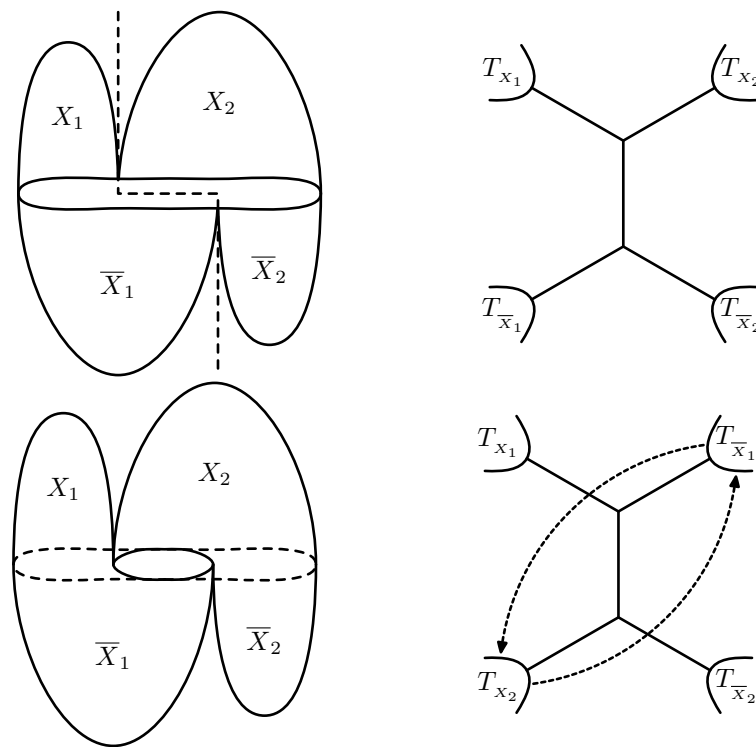


FIG. 6.4 – Schéma de la preuve de la propriété 6.25

□ Soit T_{θ_τ} la décomposition arborescente de G induite par T_τ . Montrons par l'absurde qu'un séparateur minimal de H_{T_τ} est un séparateur minimal de G .

Supposons donc qu'un séparateur minimal S de H_{T_τ} ne soit pas un séparateur minimal de G . D'après le théorème 2.28, il existe deux cliques maximales Ω_1 et Ω_2 de H_{T_τ} dont S est l'intersection. Soient v_1 et v_2 deux nœuds de T dont les étiquettes respectives $\theta_\tau(v_1)$ et $\theta_\tau(v_2)$ sont Ω_1 et Ω_2 . Il existe une arête e sur la chaîne de v_1 à v_2 qui correspond à la frontière S . Soient *T_1 et *T_2 les branches de $T \setminus \{e\}$ contenant respectivement les nœuds v_1 et v_2 et X_1 et X_2 leurs supports. Comme S n'est pas un séparateur minimal de G , $G \setminus S$ a au plus une composante connexe pleine. Nous pouvons supposer qu'aucun paquet de X_1 n'a pour voisinage S . Nous allons montrer qu'une frontière qui borde v_1 croise S et donc que T_τ n'est pas parallèle.

Soient S_1 , S_2 et S_3 les trois frontières qui bordent v_1 , P un paquet de X_1 qui capture un sommet x de $\Omega_1 \setminus S$ et y un sommet de S qui n'est pas capturé par P . Comme d'après le lemme 4.19, chaque sommet de Ω_1 apparaît dans au moins deux des trois frontières S_i , il en existe une qui contient les deux sommets x et y . Cette frontière croise S , ce qui contredit le fait que T_τ soit une décomposition parallèle.

Pour montrer qu'un séparateur minimal de H_{T_τ} induit les mêmes composantes connexes de H_{T_τ} et de G , nous allons essentiellement procéder de la même façon en raisonnant par l'absurde. Si un séparateur minimal S de H_{T_τ} n'induit pas les mêmes composantes connexes de H_{T_τ} et de G , alors il existe une clique de H_{T_τ} qui intersecte plusieurs composantes connexes de $G \setminus S$. Cette clique correspond à l'étiquette $\theta_\tau(u)$ d'un nœud de T et au moins une des frontières correspondant aux arêtes incidentes à u intersecte deux composantes connexes de $G \setminus S$. Cette frontière croise S , ce qui est absurde. ■

Des théorèmes 6.22 et 6.27 nous déduisons le théorème suivant :

Théorème 6.28 (Théorème de décomposition serrée)

Toute décomposition en branches de profil minimal induit une triangulation serrée.

Ce théorème nous permet d'appliquer l'algorithme `bw_elimination` pour décider la largeur de branches d'un hyper-graphe G et d'en déduire le théorème suivant :

Théorème 6.29

Si \mathcal{F} est une famille complète de blocs d'un hyper-graphe G de taille polynomiale telle qu'il existe une décomposition en branches de largeur minimale qui n'utilise que des blocs de \mathcal{F} et si nous savons calculer la largeur de branches de chacun des blocs de \mathcal{F} en temps polynomial, alors nous pouvons calculer la largeur de branches de G en temps polynomial.

Chapitre 7 Application à quelques classes de graphes

Sommaire

7.1	Graphes de nombre astéroïde borné	85
7.2	Graphes d'intervalles circulaires	87
7.3	Autres classes de graphes	93

Dans ce chapitre, nous appliquons le théorème 6.29 afin de calculer la largeur de branches de certaines classes de graphes : les graphes de nombre astéroïde borné ayant un nombre polynomial de séparateurs minimaux et les graphes d -trapézoïdes circulaires. Les premiers contiennent en particulier les graphes d'intervalles, les graphes de permutation et les graphes d -trapézoïdes ; les seconds étendent les graphes d'intervalles circulaires et les graphes de permutation circulaire. Ceci nous permet de retrouver les résultats de Kloks et col. [KKM99] sur le calcul de la largeur de branches des graphes d'intervalles et de les étendre. La méthode que nous utilisons pour calculer la largeur de branches des graphes d -trapézoïdes circulaires est inspirée de celle utilisée pour le calcul de la largeur arborescente des graphes d'intervalles circulaires [SSP94]. C'est pourquoi nous la présentons sur ces mêmes graphes d'intervalles circulaires avant d'expliquer comment la généraliser aux graphes d -trapézoïdes circulaires.

7.1 Graphes de nombre astéroïde borné

Broersma et col. [BKMM02] introduisent le nombre astéroïde et les ensembles astéroïdes pour étendre les graphes sans triplets astéroïdes définis par Lekkerkerker et Boland [LB62]. Dans cet article, ils démontrent que le calcul de la largeur arborescente est polynomial pour les graphes de nombre astéroïde borné avec un nombre polynomial de séparateurs minimaux.

Définition 7.1 (Ensemble astéroïde, nombre astéroïde)

Soit $G = (V, E)$ un hyper-graphe. Un ensemble A est un ensemble astéroïde si pour tout sommet a de A , les sommets de $A \setminus \{a\}$ sont dans une même composante connexe de $G \setminus N(a)$. Le nombre astéroïde $na(G)$ de G est la taille maximale d'un de ses ensembles astéroïdes.

Les blocs de ces graphes sont très particuliers. Ils sont bordés par au plus k séparateurs minimaux maximaux pour l'inclusion. Ceci nous assure qu'il y a un nombre polynomial de blocs que nous pouvons énumérer en temps polynomial.

Propriété 7.2 *Soient $G = (V, E)$ un hyper-graphe ayant un nombre astéroïde égal à k et Ω un bloc de G . L'ensemble $\Gamma_G(\Omega)$ contient au plus k éléments maximaux pour l'inclusion.*

□ Supposons par l'absurde que $\Gamma_G(\Omega)$ contienne $k + 1$ éléments maximaux pour l'inclusion. Notons les S_1, \dots, S_{k+1} . Les séparateurs minimaux S_i forment une famille de séparateurs voisins. Pour chaque indice i , il existe une composante connexe C_i^Ω de $C_G^*(S_i)$ telle que Ω soit inclus dans $C_i^\Omega \cup S_i$. Soient C_i une autre composante pleine de $C_G^*(S_i)$ et x_i un sommet de C_i . La composante C_i n'intersecte pas Ω . Montrons que $\{x_1, \dots, x_{k+1}\}$ est un ensemble astéroïde A .

Fixons un sommet x_{i_0} et considérons un sommet x_j distinct de x_{i_0} . Par hypothèse, chacun des séparateurs minimaux S_1, \dots, S_{k+1} est maximal pour l'inclusion, il existe donc un sommet y dans $S_j \setminus S_{i_0}$ et une chaîne $\mu_{x_j \rightarrow y}$ de x_j à y dans $C_j \cup \{y\}$. La composante C_j ne rencontrant pas Ω , elle n'intersecte aucun S_l . Cette chaîne évite donc S_{i_0} ce qui implique que x_j et y sont dans la même composante connexe de $G \setminus S_{i_0}$ et donc que x_j appartient à $C_{i_0}^\Omega$. Le sommet x_j étant quelconque, l'ensemble $A \setminus \{x_{i_0}\}$ est inclus dans $C_{i_0}^\Omega$. De plus, les sommets de $C_{i_0}^\Omega$ évitent les voisins de x_{i_0} . L'ensemble $A \setminus \{x_{i_0}\}$ est donc inclus dans une unique composante connexe de $G \setminus N(x_{i_0})$. Le sommet x_{i_0} étant un sommet quelconque de A , l'ensemble A est bien un ensemble astéroïde. ■

De cette propriété nous déduisons deux lemmes importants. Le premier est une application directe de la propriété 7.2 et du théorème 6.13 :

Lemme 7.3 *La largeur de branches d'un bloc d'un graphe de nombre astéroïde k est calculable en temps $O(3^k)$.*

Le second lemme est la majoration grossière suivante :

Lemme 7.4 *Un hyper-graphe de nombre astéroïde borné k a au plus $O(|\Delta_G|^k + n|\Delta_G|)$ blocs.*

□ Soient G un hyper-graphe de nombre astéroïde k et Ω un bloc de G qui n'est pas de la forme $C \cup S$. Soient S_1, \dots, S_p les séparateurs minimaux bordant Ω maximaux pour l'inclusion. Pour chaque séparateur minimal T de $\Gamma(\Omega)$, notons $C(T)$ l'unique composante pleine de $G \setminus T$ telle que $C(T) \cup T$ contienne Ω . Supposons que T soit un séparateur minimal bordant Ω inclus dans S_i . Le bloc $C(T) \cup T$ contient le bloc $C(S_i) \cup S_i$. Par conséquent, l'intersection $\bigcap_{T \in \Gamma(\Omega)} C(T) \cup T$ est égale à $\mathcal{P}(\{S_1, \dots, S_p\})$. D'après la propriété

7.2, il y a au plus $k - 1$ séparateurs S_i . Il y a donc au plus $O(|\Delta_G|^k)$ blocs de G de ce type.

Les autres blocs de G sont de la forme $C \cup S$ où S est un séparateur minimal de G et C est un élément de $C_G^*(S)$. Il y a donc au plus $n|\Delta_G|$ blocs de ce type et au final, G possède bien au plus $O(|\Delta_G|^k + n|\Delta_G|)$ blocs. ■

De ces deux lemmes et du théorème 6.29, nous déduisons directement le résultat suivant :

Théorème 7.5

Nous pouvons calculer la largeur de branches des graphes de nombre astéroïde borné ayant un nombre polynomial de séparateurs minimaux en temps polynomial.

7.2 Graphes d'intervalles circulaires

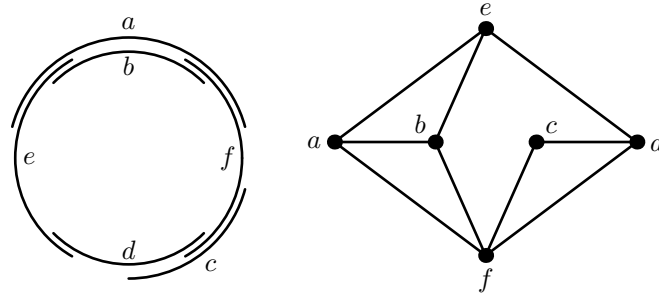
Nous nous attachons maintenant à montrer que le calcul de la largeur de branches des graphes d'intervalles circulaires est polynomial. Cependant, contrairement au cas précédent, une majoration grossière du type de celle du lemme 7.4 ne fonctionne pas car un bloc d'un graphe d'intervalle circulaire peut être bordé par un nombre arbitrairement grand de séparateurs minimaux maximaux pour l'inclusion. Nous construisons donc dans une première étape une famille \mathcal{F} de blocs de cardinal polynomial permettant néanmoins de construire une décomposition en branches de profil minimum. Nous montrons ensuite que le calcul de la largeur de branches des blocs de \mathcal{F} est polynomial et ainsi que les hypothèses d'application du théorème 6.29 sont vérifiées ; le calcul de la largeur de branches des graphes d'intervalles circulaires est donc polynomial.

Définition 7.6 (Graphes d'intervalles circulaires)

Les graphes d'intervalles circulaires sont les graphes d'intersections des arcs d'un cercle (voir fig. 7.1).

Ces graphes sont reconnaissables en temps linéaire [McC03]. De plus, l'algorithme de reconnaissance construit un modèle d'intersections. Deux modèles circulaires peuvent donner le même graphe. Nous pouvons donc choisir un modèle particulier. Par exemple, il est possible, en déplaçant légèrement les extrémités des arcs, de transformer un modèle d'intersections en un nouveau qui induit le même graphe mais dont les extrémités des arcs sont toutes distinctes. Par la suite, chaque fois que nous utilisons un graphe d'intervalles circulaires, nous supposons qu'un modèle circulaire de ce type est donné.

La largeur arborescente des graphes d'intervalles circulaires se calcule en temps $O(n^3)$ [SSP94]. Pour cela, Sundaram et col. utilisent la notion de *corde de contrôle* introduite dans [BKK95]. Cette notion de corde est la notion



À chaque arc de cercle correspond un sommet du graphe et deux sommets sont adjacents si et seulement si leurs arcs s'intersectent.

FIG. 7.1 – Un graphe d'intervalles circulaires et un de ses modèles circulaires

géométrique de corde d'un cercle. Elle est donc totalement distincte de la notion de corde qui apparaît dans la définition d'hyper-graphe triangulé.

Définition 7.7 (Point de contrôle, corde de contrôle)

Soit $\mathcal{D}(G)$ un modèle d'un graphe d'intervalles circulaires G . Entre deux extrémités consécutives d'arcs du modèle, nous plaçons un point de contrôle. Une corde reliant deux tels points est une corde de contrôle λ . La corde λ réalise l'ensemble $V(\lambda)$ des sommets correspondant aux arcs qu'elle coupe.

Les cordes de contrôle permettent de réaliser certains ensembles de sommets.

Définition 7.8 (Corde de contrôle proche d'une composante)

Soient S un séparateur de G et C une composante connexe de $G \setminus S$. La réunion des arcs correspondant aux sommets de C est un arc de cercle. Cet arc est « encadré » par deux points de contrôle. Ces points de contrôle définissent la corde de contrôle proche de C , nous la notons λ_C .

Par construction, un arc traversant la corde de contrôle λ_C proche de C correspond à un sommet qui n'appartient pas à C mais qui est adjacent à un sommet de C . L'ensemble $V(\lambda_C)$ est donc inclus dans le voisinage de C et dans S . La réciproque n'est pas vraie. Par exemple, dans la figure 7.1, l'arc b n'intersecte pas la corde proche du sommet a . Il est facile de se rendre compte qu'aucune corde ou famille de cordes de contrôle ne peut réaliser le voisinage de a . Cependant, la propriété suivante montre que les séparateurs minimaux sont les réalisations de certaines cordes de contrôle.

Propriété 7.9 Soient $\mathcal{D}(G)$ un modèle d'intersections d'un graphe d'intervalles circulaires G et C une composante pleine d'un séparateur minimal S de G . La corde proche de C réalise l'ensemble S .

□ Soient λ_C la corde proche de C et D une seconde composante pleine de S . Comme nous l'avons déjà noté, l'ensemble des sommets que réalise λ_C est inclus dans le voisinage de C . Ces sommets forment un sous-ensemble de S .

Comme aucun sommet de C n'est adjacent à un sommet de D , l'arc correspondant aux sommets de D est disjoint de celui correspondant aux sommets de C . Ainsi si un sommet appartient à la fois au voisinage de C et à celui de D , l'arc correspondant traverse λ_C . L'ensemble S est donc inclus dans $V(\lambda_C)$ ce qui achève la démonstration. ■

Ceci permet d'obtenir une majoration du nombre de séparateurs minimaux d'un graphe d'intervalles circulaires et du nombre de blocs bordés par un seul séparateur minimal maximal pour l'inclusion.

Corollaire 7.10 *Dans un graphe d'intervalles circulaires, il y a au plus $n(2n - 1)$ séparateurs minimaux et au plus $n^2(2n - 1)$ blocs bordés par un unique séparateur minimal maximal pour l'inclusion.*

Pour borner le nombre de blocs d'un graphe d'intervalles circulaires, il nous reste donc à borner le nombre de blocs qui correspondent à des parties entre séparateurs voisins. Les cordes de contrôle proches d'une composante connexe permettent aussi de représenter ces blocs. En effet, si un bloc Ω est bordé par au moins deux séparateurs minimaux maximaux pour l'inclusion, il est entièrement caractérisé par ceux-ci. La famille des cordes de contrôle Λ associée caractérise donc entièrement ce bloc. Ceci nous permet de définir la notion de représentation d'un bloc :

Définition 7.11 (Représentation d'un bloc)

Une représentation d'un bloc Ω d'un graphe d'intervalles circulaires est une famille de cordes de contrôle Λ telle que :

- (i) *la réunion des réalisations des cordes de Λ est exactement la réunion des séparateurs minimaux qui bordent Ω ;*
- (ii) *deux cordes de Λ ne se croisent pas ;*
- (iii) *il existe un domaine du disque privé des cordes de Λ qui est incident à toutes les cordes de Λ .*

Pour pouvoir borner le nombre de blocs correspondant à des parties entre séparateurs voisins, nous cherchons à borner le nombre de cordes de contrôle d'une représentation de Ω . Pour cela, nous introduisons la notion de représentation compatible avec une décomposition en branches.

Si nous considérons une décomposition parallèle T_τ de G et la triangulation serrée H_{T_τ} correspondante, chaque clique maximale Ω de H_{T_τ} correspond à un bloc de G . De plus, il existe un nœud x_Ω de T dont l'étiquette $\theta_\tau(x_\Omega)$ est Ω . Notons *T_1 , *T_2 et *T_3 les trois branches de $T \setminus \{x_\Omega\}$. D'après

le théorème 6.20, nous pouvons supposer que les supports des branches $(^*T_i)_{i \in \{1,2,3\}}$ ne sont pas morcelés dans T_τ . Ceci implique que les paquets correspondant aux composantes connexes de $G \setminus \Omega$ sont les supports de sous-branches $(^*T_j^i)_{j \in J_i}$ des branches $(^*T_i)_{i \in \{1,2,3\}}$. De plus, si deux paquets ont la même frontière, ils correspondent à des sous-branches d'une même branche *T_i . Comme les séparateurs minimaux qui bordent Ω sont exactement les frontières des paquets correspondant aux composantes connexes de $G \setminus \Omega$, nous pouvons les étiqueter avec le numéro de la branche *T_i dans laquelle se trouve la sous-branche $^*T_j^i$ dont ils sont la frontière. Dans ce cas, nous disons qu'un séparateur minimal qui borde Ω *appartient* à une branche *T_i .

Définition 7.12 (Représentation compatible)

Une représentation de Ω compatible avec T_τ est une représentation de Ω qui vérifie les conditions supplémentaires :

- (i) chaque séparateur minimal qui borde Ω est inclus dans la réalisation $V(\lambda)$ d'une corde de Λ ;
- (ii) les séparateurs minimaux inclus dans la réalisation d'une corde de Λ appartiennent tous à la même branche *T_i .

La première condition de la définition 7.12 est un renforcement de la première condition de la définition 7.11. Quant à la seconde condition, elle permet d'étiqueter chaque corde d'une représentation compatible comme les séparateurs minimaux qui bordent Ω .

La propriété suivante permet d'affirmer qu'il existe une représentation de Ω compatible avec T_τ .

Propriété 7.13 Soient $\mathcal{D}(G)$ un modèle d'intersections d'un graphe d'intervalles circulaires G et Ω un bloc de G . Il existe une famille de cordes Λ telle que :

- (i) les cordes de Λ réalisent les séparateurs minimaux de $\Gamma(\Omega)$;
- (ii) deux cordes de Λ ne se croisent pas ;
- (iii) il existe un domaine du disque privé des cordes de Λ qui est incident à toutes les cordes de Λ .

□ Soit Λ l'ensemble des cordes de contrôle proches des composantes connexes de $G \setminus \Omega$. Comme les séparateurs minimaux qui bordent Ω sont exactement les voisinages des composantes connexes de $G \setminus \Omega$, la propriété 7.9 permet d'affirmer que les cordes de Λ réalisent bien les éléments de $\Gamma(\Omega)$. Par construction, Λ vérifie aussi les deux autres conditions. ■

Bien que nous ne connaissions pas T_τ , si Ω est un bloc de T_τ , alors chaque séparateur minimal bordant Ω est inclus dans l'une des frontières d'une des branches *T_i . De cette façon, Λ est compatible avec n'importe quelle décomposition en branches dont Ω est un bloc.

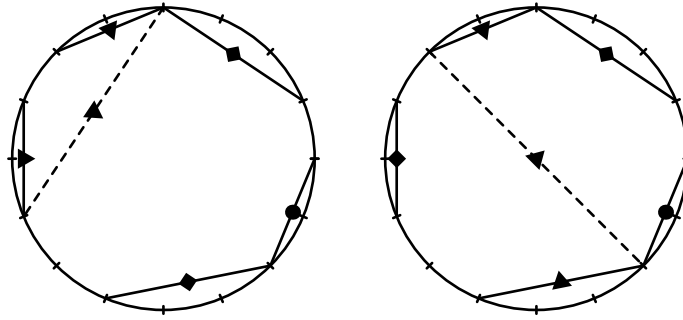
Nous sommes à même de borner le nombre de cordes d'une représentation des blocs de G correspondant à des parties entre séparateurs voisins.

Théorème 7.14

Soient T_τ une décomposition de profil minimal d'un graphe d'intervalles circulaires G et Ω un bloc de G correspondant à une clique maximale de la triangulation induite par T_τ . Une représentation Λ de Ω de cardinal minimal compatible avec T_τ contient au plus trois cordes.

□ Raisonnons par l'absurde et supposons qu'une représentation Λ de Ω de cardinal minimum compatible avec T_τ contient au moins quatre cordes de contrôle.

Il existe dans Λ au moins deux cordes λ_1 et λ_2 appartenant à une même branche. Nous pouvons supposer que cette branche est *T_1 . Les extrémités de λ_1 et λ_2 permettent de construire une diagonale λ_3 du polygone dont les sommets sont les extrémités des cordes de Λ (voir fig. 7.2).



Voici deux représentations de blocs et deux diagonales.

Un même symbole apparaît sur les cordes d'une même branche.

Dans le premier cas, la diagonale permet soit de diminuer le nombre de cordes de la représentation, soit de diminuer le profil de la décomposition. Dans le second cas, elle permet de diminuer le profil de la décomposition.

FIG. 7.2 – Preuve du théorème 7.14

Nous allons transformer T_τ de manière à respecter la structure de $\Lambda \cup \{\lambda_3\}$. La corde λ_3 partitionne Λ en deux ensembles Λ_1 et Λ_2 . Grâce au théorème 6.20, nous pouvons supposer que les supports des branches *T_1 , *T_2 et *T_3 ne sont pas morcelés. Nous pouvons donc construire six branches $({}^*T_i^j)_{i \in \{1,2,3\}, j \in \{1,2\}}$ en regroupant les branches correspondant aux paquets de $\tau({}^*T_i)$ dont la frontière est incluse dans une corde de Λ_j . En regroupant les branches ${}^*T_1^1$, ${}^*T_2^1$ et ${}^*T_3^1$, nous obtenons une branche ${}^*T_1'$. De même, nous construisons une branche ${}^*T_2'$ que nous regroupons avec ${}^*T_1'$ pour obtenir une décomposition en branches $T_{\tau'}^i$ de G . Les frontières de $\tau({}^*T_i^1)$ et de $\tau({}^*T_i^2)$ sont incluses dans celles de *T_i . De plus, la frontière de $\tau({}^*T_1')$ correspond à des

arcs qui croisent λ_3 . Or, par construction, tout arc croisant λ_3 croise λ_1 ou λ_2 . La frontière de $\tau(*T'_1)$ est donc incluse dans celle de $\tau(*T_1)$. D'après le lemme 6.19, le profil de $T'_{\tau'}$ est inférieur ou égal à celui de T_{τ} .

Plusieurs cas se présentent. Si toutes les cordes de Λ_1 appartiennent à la même frontière et si $\bigcup_{\lambda \in \Lambda_1} V(\lambda)$ est égal à $V(\lambda_3)$, alors $\Lambda_2 \cup \{\lambda_3\}$ est une représentation de Ω compatible avec T_{τ} et de cardinal strictement plus petit que celui de Λ , ce qui est absurde. La situation est la même pour Λ_2 . Sinon, il existe un sommet x de $\bigcup_{\lambda \in \Lambda_1} V(\lambda)$ qui n'appartient pas à $V(\lambda_3)$ et un sommet y de $\bigcup_{\lambda \in \Lambda_2} V(\lambda)$ qui n'appartient pas à $V(\lambda_3)$. Comme, d'après le lemme 4.19 les sommets x et y appartiennent à deux frontières de $\tau(*T_i)$, nous pouvons supposer que la frontière de $\tau(*T_2)$ croise $V(\lambda_3)$. Aucune frontière de $T'_{\tau'}$ ne contient celle de $\tau(*T_2)$. D'après le lemme 6.19, le profil de $T'_{\tau'}$ est strictement inférieur à celui de T_{τ} ce qui est aussi absurde. ■

Nous déduisons de ce théorème le corollaire suivant :

Corollaire 7.15 *Il existe au plus $O(n^6)$ blocs pouvant apparaître dans une décomposition de profil minimal. Ils sont énumérables en temps polynomial.*

□ D'après le corollaire 7.10, il y a $O(n^3)$ blocs bordés par un unique séparateur minimal maximal pour l'inclusion. Le théorème 7.14 montre que les blocs correspondant à des parties entre séparateurs minimaux sont représentables par au plus trois cordes de contrôle. Une fois les extrémités des cordes fixées, il existe un nombre borné de cordes s'appuyant sur ces points de contrôle. Il y a donc $O(n^6)$ familles d'au plus trois cordes de contrôle et $O(n^6)$ blocs correspondant à des parties entre séparateurs. ■

Nous pouvons montrer que le théorème 6.29 s'applique et donc que nous pouvons calculer la largeur de branches des graphes d'intervalles circulaires en temps polynomial.

Théorème 7.16

La largeur de branches d'un graphe d'intervalles circulaires se calcule en temps polynomial.

□ Soit G un graphe d'intervalles circulaires. Le corollaire 7.15 nous assure qu'il existe une famille complète \mathcal{F} de taille $O(n^6)$ de blocs de G . Pour appliquer le théorème 6.29, nous devons pouvoir calculer la largeur de branches de ces blocs en temps polynomial.

Soient Ω un bloc de cette famille complète et Λ_{Ω} une famille d'au plus trois cordes de contrôle réalisant Ω . La famille Λ_{Ω} donne une indication de la façon de décomposer Ω : les réalisations de chacune des cordes de Λ doivent se trouver dans une même branche de la décomposition du bloc. Cette représentation ne donne pas forcément une décomposition optimale de Ω mais, d'après le théorème 7.14 il existe une réalisation de Ω pour laquelle c'est le cas. Pour construire \mathcal{F} , il suffit donc de considérer chaque famille Λ_{Ω}

d'au moins trois cordes de contrôle correspondant à un bloc Ω et de calculer la largeur que donne Λ_Ω . Ainsi, nous obtenons une famille de blocs Ω et un majorant de la largeur de branches de Ω par réalisation de Ω . Il suffit de choisir la plus petite valeur obtenue pour Ω pour obtenir sa largeur de branches.

Il nous reste à calculer la largeur de branches correspondant à la famille Λ_Ω . Pour cela, nous devons considérer une clique sur l'ensemble de sommets Ω à laquelle nous ajoutons des hyper-arêtes correspondant à la réalisation de chacune des cordes de Λ_Ω . D'après le théorème 6.13, la largeur de branches de cette hyper-clique se calcule en temps constant.

Pour stocker les blocs intermédiaires, nous pouvons utiliser une structure d'arbre de recherche. Les différentes opérations se font en temps $\log(\text{nombre d'éléments})$ donc au plus en temps $O(\log n)$. Le temps de calcul pour chacune des familles de cordes de contrôle étant constant, le temps total de calcul de \mathcal{F} et des largeurs de branches de ses éléments est $O(n^6 \log n)$. ■

7.3 Autres classes de graphes

La méthode présentée pour calculer la largeur de branches des graphes d'intervalles circulaires peut se généraliser. Pour cela, deux conditions sont nécessaires :

- (i) il doit exister une notion de corde de contrôle qui permette de réaliser les séparateurs minimaux et qui fournisse des représentations des blocs de G ;
- (ii) la réalisation d'une diagonale doit être incluse dans les réalisations des cordes sur les extrémités desquelles elle s'appuie.

Nous pouvons ainsi appliquer cette méthode aux graphes de permutation circulaire, aux graphes trapézoïdes circulaires et aux graphes d -trapézoïdes circulaires.

Définition 7.17 (Graphes de permutation circulaire)

Considérons le tore $(\mathbb{R}/[0, n[) \times [0, 1]$ et relient de façon bijective les points $(i, 0)$ aux points $(\sigma(i), 1)$ (i entier entre 0 et $n - 1$) par des segments s_i . Un graphe de permutation circulaire est le graphe d'intersections des segments s_i .

À l'aide d'une inversion, nous pouvons aussi donner un modèle d'un graphe de permutation circulaire du type de celui de la figure 7.3.

Définition 7.18 (Graphes trapézoïdes circulaires)

Les graphes trapézoïdes circulaires sont les graphes d'intersections d'une famille de trapèzes du tore $(\mathbb{R}/[0, 1[) \times [0, a]$ ($a \geq 0$) dont les bases sont sur les droites $(\mathbb{R}/[0, 1[) \times \{0\}$ et $(\mathbb{R}/[0, 1[) \times \{a\}$.

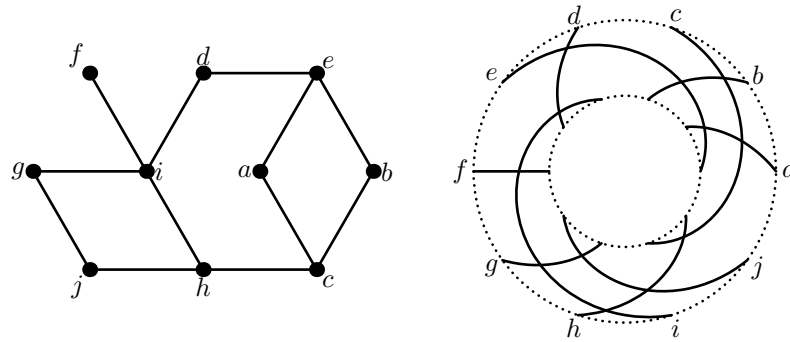


FIG. 7.3 – Un graphe de permutation circulaire et son modèle

À l'aide d'une inversion, nous pouvons aussi donner un modèle d'un graphe de permutation circulaire du type de celui de la figure 7.4.

Les graphes trapézoïdes circulaires sont une sur-classe des graphes d'intervalles circulaires et des graphes de permutation circulaire. Pour obtenir les graphes d'intervalles circulaires, il suffit de prendre des trapèzes d'épaisseur nulle et pour obtenir les graphes de permutation circulaire, il suffit de choisir des trapèzes dont les bases sont de taille nulle.

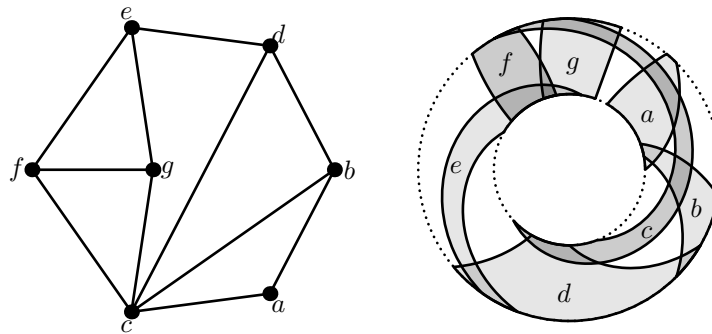


FIG. 7.4 – Un graphe trapézoïde circulaire et son modèle

Définition 7.19 (Graphes d -trapézoïdes circulaires)

Les graphes d -trapézoïdes circulaires sont les graphes d'intersections d'une famille d'empilements de d trapèzes du tore $(\mathbb{R}/[0,1[) \times [0, a_d]$ tels que les bases du p -ième trapèze soient sur les droites $(\mathbb{R}/[0,1[) \times \{a_p\}$ et $(\mathbb{R}/[0,1[) \times \{a_{p+1}\}$ ($0 \leq a_1 \cdots \leq a_d$) et tels que la base inférieure du p -ième trapèze soit la base supérieure du $(p-1)$ -ième trapèze.

Les graphes de cordes forment une classe de graphes pour laquelle nous n'avons pas réussi à adapter cette méthode. En effet, la seconde condition nécessaire à l'application de cette méthode n'est pas remplie.

Définition 7.20 (Graphes de cordes)

Les graphes de cordes sont les graphes d'intersections de cordes d'un cercle (voir fig. 7.5).

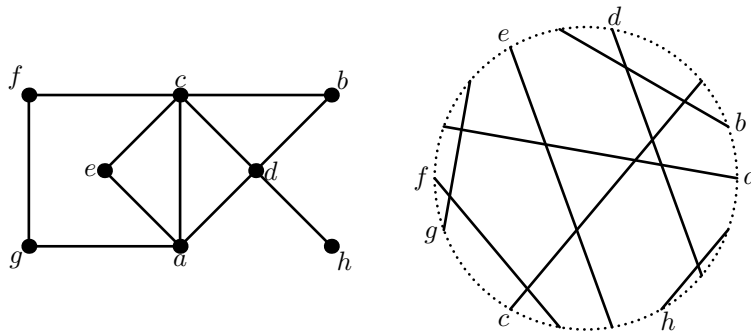


FIG. 7.5 – Un graphe de cordes et son modèle

Cependant, en utilisant d'autres arguments que ceux présentés ici, nous pouvons espérer arriver à borner le nombre de cordes de contrôle d'une représentation d'un bloc. Les travaux de L. Lyaudet [Lya04] comportent une tentative dans ce sens. Dans [KKM99], Kloks et col. montrent que le problème de la largeur de branches est NP-complet pour les split-graphes. Pour cela, ils montrent que si (C, I, E) est un split-graphe, décider si la largeur de branches du graphe vaut $\lceil 3|C|/N \rceil$ est NP-complet. Lyaudet montre que ce problème est linéaire pour les split-graphes de cordes. Sa démonstration permet d'espérer pouvoir étendre notre démarche aux graphes de cordes.

*La géométrie est l'art de raisonner juste
sur des figures fausses.*

Folklore

Sommaire

8.1 Séparateurs, séparations et graphe intermédiaire . . .	99
8.1.1 Topologie des graphes et hyper-graphes planaires . . .	99
8.1.2 Courbes de Jordan et séparations	102
8.1.3 Homotopie et graphe intermédiaire	103
8.1.4 Séparateurs minimaux des graphes planaires . . .	110
8.2 Énumération des séparateurs minimaux	115
8.3 Représentations de décompositions	119
8.4 Dualité et largeur arborescente	122
8.4.1 Définition et théorème de dualité	122
8.4.2 Dualité et graphe intermédiaire	125
8.4.3 Deux théorèmes de dualité	126

Historiquement, la théorie des graphes est née avec les graphes planaires dans les travaux d'Euler. Ces graphes dessinables sur le plan ou la sphère possèdent des propriétés spécifiques liées à la topologies des surfaces dans lesquelles ils sont plongés. Nous disposons donc pour les étudier d'outils puissants de topologie plane qui, combinés aux résultats que nous avons déjà obtenus, nous permettent d'obtenir des résultats spécifiques à ces graphes.

Avant d'aborder les résultats et les preuves formelles qui sont émaillés de propriétés topologiques faussement triviales, voici une présentation informelle les grandes lignes de ce chapitre.

Pour construire une séparation d'un graphe planaire G , nous pouvons tracer une courbe qui ne coupe aucune arête de G et qui délimite deux régions du plan ; ces courbes sont des courbes de Jordan. De cette façon, les deux régions définissent une partition des arêtes de G . Si toutes les séparations de G ne peuvent pas être représentées de cette façon, nous montrons que les séparateurs minimaux peuvent toujours l'être. À l'inverse, en déformant un peu une courbe associée à une séparation, nous obtenons une nouvelle courbe représentant la même séparation : ceci nous conduit à introduire la notion d'homotopie qui formalise ces déformations. Une courbe

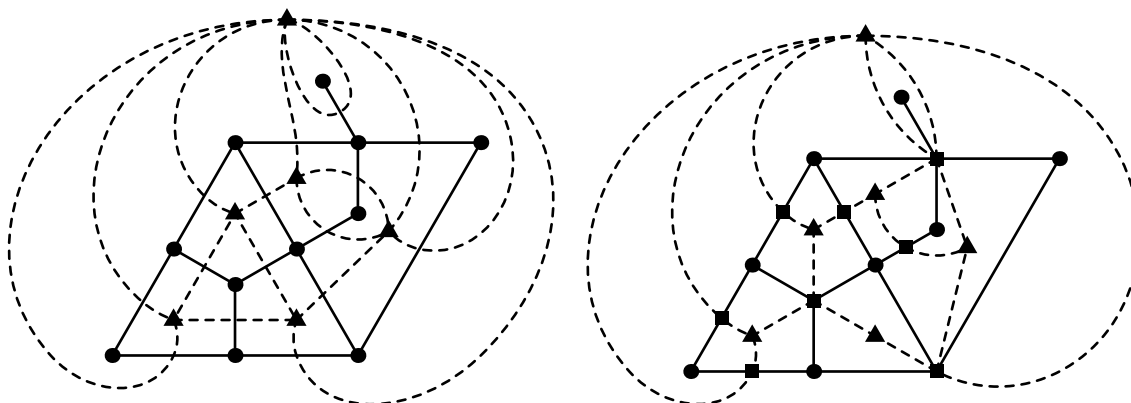
représentant une séparation donnée passe alternativement par des sommets de G puis par des faces de G . De plus, deux courbes associées à la même suite de sommets et de faces définissent la même séparation. Ainsi, en fixant un sommet dans chaque face et en le reliant à tous les sommets bordant cette face, nous obtenons un graphe appelé graphe intermédiaire G_I . Les cycles élémentaires de ce graphe représentent donc des séparations de G et en particulier les séparateurs minimaux de G . Ceci nous permet d'énumérer plus rapidement les séparateurs minimaux d'un graphe planaire.

Si deux cycles élémentaires de G_I ne se croisent pas, les deux séparations qu'ils induisent ne se chevauchent pas. Ainsi une famille maximale de cycles élémentaires de G_I définit une matroïche dont nous montrons qu'elle est complète. De cette façon, nous pouvons définir des décompositions en branches dont certaines correspondant à des décompositions arborescentes de largeur minimale. Les nœuds de ces décompositions correspondent à des Θ -structures, c'est-à-dire à des ensembles de trois cycles qui dessinent des Θ . En remarquant qu'un graphe planaire et son graphe dual ont le même graphe intermédiaire, nous pouvons transposer une bonne décomposition arborescente de G de largeur minimale à son dual G^* . Chaque Θ -structure que définit la famille de cycles correspond à la fois à un nœud de la décomposition arborescente de G et à un nœud de celle de G^* . De cette façon, nous pouvons majorer finement la largeur arborescente de la décomposition duale ce qui nous permet d'affirmer que la largeur arborescente d'un graphe planaire et celle de son dual diffèrent d'au plus un.

Conventions graphiques et typographiques

Les preuves formelles des résultats que nous venons d'évoquer s'avèrent bien souvent délicates même si les idées qui les sous-tendent sont relativement simples. Nous utilisons donc de nombreux dessins pour illustrer notre propos. Nous avons évoqué dans notre présentation du graphe intermédiaire différents types de sommets et pour faciliter la lecture des dessins tout en allégeant les légendes, nous représentons les différents types de sommets par des figurés différents résumés dans la figure 8.1.

Les dessins facilitent la lecture des preuves ; néanmoins, ils ne constituent pas une preuve et peuvent même induire en erreur en masquant certaines difficultés. La topologie plane regorge d'évidences non triviales comme le théorème de Jordan (th 8.8). Par souci de rigueur, nous démontrons tous nos résultats, mais certaines des preuves sont avant tout techniques et n'apportent que peu de choses à la compréhension générale du problème. Pour ne pas trop masquer le fil conducteur et ne pas perdre le lecteur au milieu de ces démonstrations, nous utilisons la convention typographique suivante



Les sommets initiaux sont représentés par des ronds,
 les sommets arêtes par des carrés,
 les sommets faces par des triangles.

Dans le cas d'un graphe simple, pour ne pas surcharger le
 dessin, nous ne représentons pas les sommets arêtes.

FIG. 8.1 – Conventions graphiques

pour présenter les preuves et les lemmes techniques.

Lemme 154.17 Ceci est un lemme « évident ».

□ Cependant la preuve est pénible. ■

8.1 Séparateurs, séparations et graphe intermédiaire

8.1.1 Topologie des graphes et hyper-graphes planaires

Nous définissons ici les graphes et hyper-graphes plans et nous introduisons les premiers outils topologiques dont nous aurons besoin.

Définition 8.1 (Chemin, lacet)

Un chemin entre deux points x et y d'un espace métrique X est une fonction $f : [0, 1] \mapsto X$ continue, dérivable par morceaux telle que $f(0) = x$, $f(1) = y$. Un lacet est un chemin dont les deux extrémités sont identiques.

Le chemin f^{-1} désigne le chemin f « pris à l'envers », c'est-à-dire la fonction f^{-1} qui à t associe $f(1 - t)$. Si ν et μ sont deux chemins tels que $\nu(1)$ soit égal à $\mu(0)$, $\nu.\mu$ désigne la concaténation de ν et μ .

Définitions 8.2 (Graphe plan, hyper-graphe plan, face)

Un graphe plan est un couple $G_\Sigma = (V_\Sigma, E_\Sigma)$ d'ensembles finis. Les éléments de V_Σ sont appelés les sommets et ceux de E_Σ les arêtes. Les ensembles V_Σ et E_Σ sont tels que :

- $V_\Sigma \subseteq \Sigma$;
- chaque arête e de E_Σ est un chemin dont les extrémités sont des sommets;
- l'intérieur d'une arête ne contient aucun sommet ni aucun point d'une autre arête.

Un hyper-graphe plan est un graphe plan biparti $G_\Sigma = (V_\Sigma \cup A_\Sigma, E_\Sigma)$. Les éléments de V_Σ sont les sommets initiaux et ceux de A_Σ sont les sommets arêtes de G_Σ .

La sphère Σ privée du dessin de G_Σ est un ouvert. Ses composantes connexes sont des ouverts connexes par arcs : les faces du graphe ou de l'hyper-graphe plan. Pour chaque face f de G , nous choisissons un point x de f comme représentant de la face que nous appelons centre de la face. Nous parlons indifféremment du centre x d'une face pour désigner le point x ou pour désigner la face qu'il représente.

Définition 8.3 (Graphe planaire, hyper-graphe planaire)

À un graphe plan correspond naturellement un multi-graphe abstrait G . Réciproquement, un multi-graphe abstrait G est dit planaire s'il existe un graphe plan G_Σ dont il est le multi-graphe abstrait. Un tel graphe plan est une représentation plane de G .

À un hyper-graphe plan correspond naturellement le graphe d'incidence d'un hyper-graphe et donc un hyper-graphe. Réciproquement, un hyper-graphe G est dit planaire s'il existe un hyper-graphe plan G_Σ dont il est l'hyper-graphe abstrait. Un tel hyper-graphe plan est une représentation plane de G .

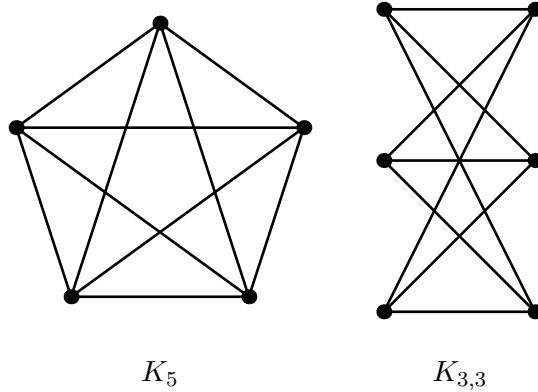
Deux graphes plan peuvent induire le même graphe abstrait. C'est en particulier le cas de deux graphes plans dont les dessins sont homéomorphes, c'est pourquoi nous raisonnerons toujours par la suite à homéomorphisme près. Cependant, même comme cela, un graphe planaire peut avoir deux représentations planes non homéomorphes. En revanche, un graphe planaire 3-connexe admet une unique représentation plane.

Il est relativement facile de se convaincre que les hyper-graphes planaires sont clos par les opérations suivantes : suppression d'un sommet, suppression d'une hyper-arête, contraction d'une hyper-arête e en un nouveau sommet v_e dont les voisins sont ceux des extrémités de e . Une classe d'hyper-graphe stable par ces propriétés est *close par minoration*. En utilisant la relation d'Euler entre les nombres de sommets, de faces et d'arêtes d'un graphe plan, il est relativement facile de montrer que les graphes K_5 et $K_{3,3}$ (voir fig. 8.2) ne sont pas planaires. Nous renvoyons le lecteur à la lecture de [Die00] pour de plus amples détails. Ainsi, un graphe qui contient K_5 ou $K_{3,3}$ comme

mineur n'est pas planaire. Le théorème de Kuratowski donne une réciproque à cette propriété.

Théorème 8.4 (Kuratowski 1930)

Un graphe est planaire si et seulement si il ne contient pas de mineur isomorphe à K_5 ou $K_{3,3}$.



Un graphe est planaire si et seulement si il n'admet pas l'un des deux graphes ci-dessus comme mineur.

FIG. 8.2 – Théorème de Kuratowski (th. 8.4)

Nous avons vu qu'en général, les séparateurs minimaux d'un hyper-graphe n'étaient pas minimaux pour l'inclusion. Ce n'est toujours pas vrai pour les hyper-graphes planaires mais le théorème de Kuratowski permet de montrer que c'est presque le cas, ce qui simplifie considérablement la tâche pour énumérer les séparateurs minimaux d'un hyper-graphe planaire.

Propriété 8.5 *Soient G un hyper-graphe planaire et S_1 et S_2 deux séparateurs minimaux distincts de G . Si S_1 est inclus dans S_2 , alors S_1 contient au plus deux sommets.*

□ Supposons par l'absurde que S_1 soit strictement inclus dans S_2 et qu'il contienne au moins trois éléments α, β et γ . Soient a et b deux sommets tels que S_2 soit un a, b -séparateur minimal. Comme S_1 est strictement inclus dans S_2 , les sommets a et b sont dans la même composante pleine $C_G^a(S_1)$. Comme S_1 est un séparateur minimal, il admet une autre composante pleine. Soit c dans une telle composante. Le séparateur S_1 est un a, c -séparateur minimal. Contractons les composantes $C_G^a(S_2)$, $C_G^b(S_2)$ et $C_G^c(S_1)$ en trois sommets a', b' et c' . Chacun de ces trois sommets est adjacent à S_1 donc à α, β et γ . L'hyper-graphe contient donc un mineur isomorphe à $K_{3,3}$ en contradiction avec le théorème de Kuratowski. ■

8.1.2 Courbes de Jordan et séparations

Nous avons utilisé au chapitre 7 certaines frontières ayant une interprétation géométrique simple. Ce faisant, nous avons pu obtenir des propriétés supplémentaires et ainsi calculer la largeur de branches de certaines classes de graphes. Les hyper-graphes planaires ont une structure topologique forte comme l'illustre le théorème de Kuratowski. Il est donc naturel de suivre une démarche analogue et de restreindre la forme des frontières que nous allons utiliser dans le cas des hyper-graphes planaires.

Définition 8.6 (Chemin compatible avec un hyper-graphe plan)

Un chemin μ est compatible avec un graphe ou un hyper-graphe plan G_Σ si μ n'intersecte que des sommets initiaux de G_Σ . Nous notons $V_{G_\Sigma}(\mu)$ l'ensemble $\mu \cap V_\Sigma$ des sommets initiaux de G_Σ que rencontre μ . Si nous notons Σ_{G_Σ} la sphère privée des sommets arêtes et de l'intérieur des arêtes de G_Σ , un chemin compatible avec G_Σ est un chemin de Σ_{G_Σ} .

Définition 8.7 (Courbe de Jordan)

Une courbe de Jordan est un lacet μ de Σ tel que si $\mu(t_1) = \mu(t_2)$ et $t_1 < t_2$ alors $t_1 = 0$ et $t_2 = 1$. Si un lacet n'est pas une courbe de Jordan, il est singulier.

Deux courbes de Jordan μ et ν sont parallèles si μ n'intersecte qu'une composante connexe de $\Sigma \setminus \nu$ et réciproquement.

Théorème 8.8 (Jordan)

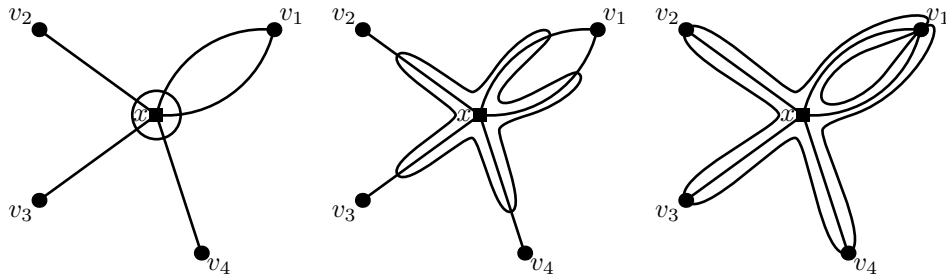
Soit μ une courbe de Jordan de Σ . L'ensemble $\Sigma \setminus \mu$ est constitué de deux composantes connexes homéomorphes au disque unité ouvert.

Le théorème de Jordan montre qu'une courbe de Jordan partage la sphère en deux. Ainsi, une courbe de Jordan μ compatible avec un hyper-graphe plan G_Σ induit une partition éventuellement dégénérée des arêtes de l'hyper-graphe associé G . Il est donc naturel de se demander si nous ne pourrions pas utiliser seulement des décompositions en branches de G dont toutes les séparations sont induites par des courbes de Jordan compatibles avec G_Σ .

Lemme 8.9 Soient $G_\Sigma = (V_\Sigma, E_\Sigma)$ un graphe plan et x dans V_Σ . Il existe un lacet μ compatible avec G_Σ tel que :

- (i) $V_{G_\Sigma}(\mu) = N(x)$;
- (ii) il existe une composante connexe C de $\Sigma_{G_\Sigma} \setminus \mu$ qui ne contient que le sommet x .

□ Soit x dans V_Σ . Considérons un cercle de centre x qui sépare x des autres sommets de V_Σ . Ce cercle coupe les arêtes $(e_i)_{i \in I}$ de G_Σ incidentes à x . En déformant ce cercle au niveau de ses intersections avec les arêtes de G_Σ , nous pouvons obtenir un lacet μ qui passe par les autres extrémités des arêtes e_i et qui sépare encore x des autres sommets de G_Σ (figure 8.3). ■

FIG. 8.3 – Obtention du lacet μ

En utilisant ce lemme, nous montrons (prop. 8.19) que les séparateurs minimaux peuvent être vues comme des courbes de Jordan.

8.1.3 Homotopie et graphe intermédiaire

Utiliser des courbes de Jordan pour représenter des séparations est séduisant car cela permet de donner une représentation visuelle de ces séparations. Cependant, le problème majeur de cette approche est que deux courbes de Jordan peuvent représenter la même séparation comme l'illustre la figure 8.4. De plus, si nous voulons étendre l'algorithme 3.3 en utilisant cette idée, nous devons nous ramener à une structure combinatoire simple pour représenter de façon efficace des courbes de Jordan.

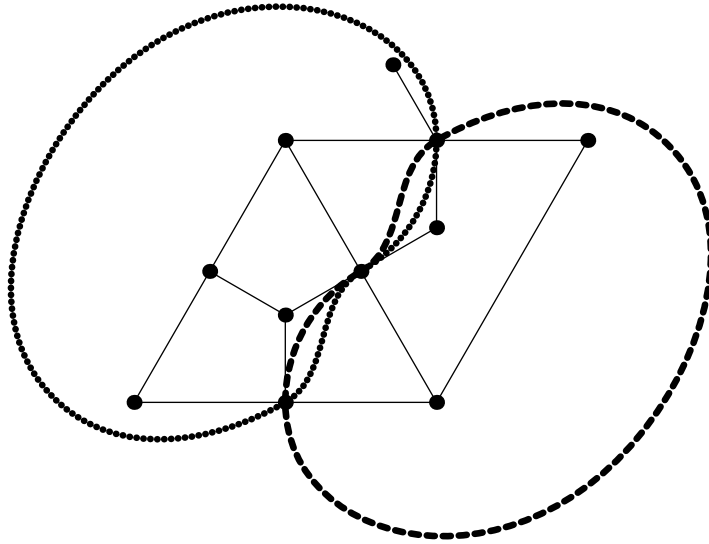


FIG. 8.4 – Deux courbes de Jordan induisant une même séparation

Revenons à la figure 8.4. Au cours d'une déformation compatible avec G_Σ , une courbe de Jordan ne peut pas « sauter par dessus » une arête. Une courbe de Jordan obtenue en transformant ainsi une courbe μ compatible

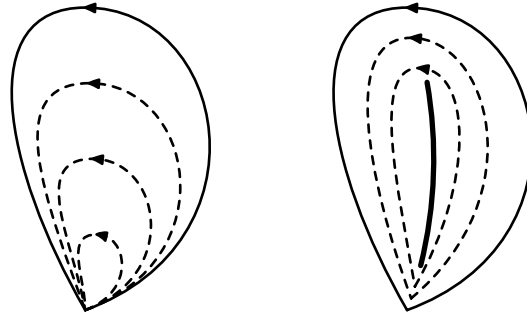
avec G_Σ induit la même séparation que μ . Nous pouvons ainsi définir une relation d'équivalence sur les courbes de Jordan compatibles avec G_Σ et raisonner modulo cette relation. Pour formaliser cette idée de déformation, nous utilisons la notion d'homotopie.

Définition 8.10 (Homotopie)

Deux lacets ν et μ sont dits homotopes dans un espace métrique A s'ils appartiennent à A et si il est possible de passer de façon continue de ν à μ dans A , c'est-à-dire s'il existe $F: [0, 1] \times [0, 1] \mapsto A$ continue telle que :

- à t fixé dans $[0, 1]$, $x \rightarrow F(x, t)$ soit un lacet ;
- le chemin $x \rightarrow F(x, 0)$ soit égal à ν ;
- le chemin $x \rightarrow F(x, 1)$ soit égal à μ .

Un lacet est homotope à zéro s'il est homotope au lacet constant (cf. fig. 8.5). Deux chemins ν et μ sont dits homotopes s'ils ont les mêmes extrémités et si $\nu \cdot (\mu^{-1})$ est homotope à zéro (cf. fig. 8.6). La relation d'homotopie sur les lacets et sur les chemins est une relation d'équivalence.



Le premier lacet est dans le plan. Il est homotope à zéro.
Le second lacet est dans le plan privé du trait gras. Il n'est pas homotope à zéro.

FIG. 8.5 – Homotopie de lacets dans le plan

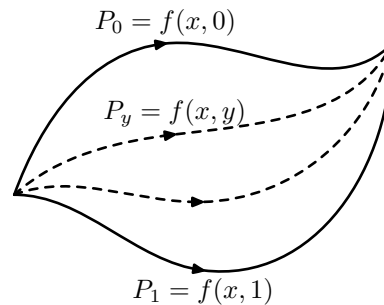


FIG. 8.6 – Homotopie de chemins

Même si nous pouvons facilement plonger un hyper-graphe plan de la sphère dans le plan et inversement, nous devons tout de même choisir une face extérieure. Ce choix n'est certainement pas anodin. En effet, cela revient à « trouser » la face extérieure. Ainsi, deux lacets qui ne sont pas homotopes dans le plan (cf. fig. 8.5) peuvent être issus de chemins homotopes sur la sphère. Le second lacet de la figure 8.5 correspond à celui de la figure 8.7 qui est bien homotope à zéro. La figure 8.4 fournit un autre exemple de cette différence : les deux courbes de Jordan représentées ne sont pas homotopes dans le plan privé de l'intérieur des arêtes du graphe alors qu'elles le sont si nous nous plaçons sur la sphère.

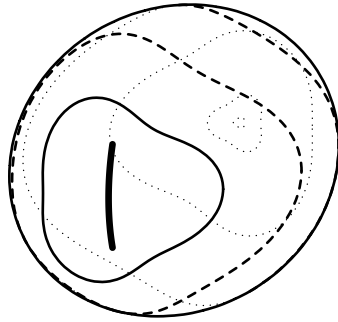


FIG. 8.7 – Homotopie de lacets sur la sphère

L'utilisation de l'homotopie pose également un problème technique : le choix du paramétrage. Les images d'un lacet μ et du lacet μ^{-1} sont les mêmes – le second « fait le tour à l'envers » – mais ils ne sont pas nécessairement homotopes. Pour éviter ce type de problèmes, nous dirons que deux chemins sont homotopes s'il existe des paramétrages de leurs images qui le soit. Une fois cette précaution prise, nous pouvons énoncer la propriété suivante :

Propriété 8.11 *Deux courbes de Jordan μ et ν de Σ compatibles avec un hyper-graphe plan G_Σ induisent les mêmes séparations de G si et seulement si elles sont homotopes dans Σ_{G_Σ} .*

La propriété 8.11 permet de raisonner sur des classes d'équivalence de courbes de Jordan de Σ_{G_Σ} pour la relation d'homotopie. Nous allons montrer comment ce passage au quotient nous permet de définir un graphe plan.

Soit μ une courbe de Jordan compatible avec un hyper-graphe plan G_Σ . Nous pouvons décomposer μ en une somme de chemins élémentaires $\mu_1 \cdot \mu_2 \dots \mu_p$ de telle sorte que chaque chemin μ_i intersecte le dessin de G_Σ au niveau de ses extrémités et uniquement à cet endroit. Comme une face est simplement connexe, chaque μ_i est homotope à un chemin qui passe par le centre d'une face de G_Σ . La courbe μ se décompose donc en $\mu_1 \cdot \nu_1 \dots \mu_p \cdot \nu_p$ où chaque μ_i relie un sommet au centre d'une face de G_Σ et chaque ν_i relie le centre d'une face à un sommet de G_Σ .

Définition 8.12 (Graphe intermédiaire)

Soient $G_\Sigma = (V_\Sigma \cup A_\Sigma, E_\Sigma)$ un hyper-graphe plan et F_Σ l'ensemble des centres des faces de G_Σ .

Nous construisons un graphe plan biparti G_I ayant comme sommets les sommets initiaux et les centres des faces de G_Σ . Pour chaque chemin μ dont l'intérieur est inclus dans une face F de G_Σ et qui relie le centre f de F et un sommet initial v incident à F , nous plaçons une arête dans G_I homotope à μ .

Il est possible de choisir les arêtes de G_I pour obtenir un graphe plan. Un tel graphe est unique à homéomorphisme près ; c'est le graphe intermédiaire de G_Σ . Les éléments de F_Σ sont appelés sommets faces.

La figure 8.8 présente un graphe plan et son graphe intermédiaire.

Nous pouvons aussi remarquer que si x n'est pas un point d'articulation de G – c'est-à-dire un séparateur de taille un –, pour chaque face f à laquelle x est incident, il y a une seule arête dans E_I . Dans le cas d'un hyper-graphe plan 2-connexe, la définition de graphe intermédiaire peut être simplifiée :

Définition 8.13 (Graphe intermédiaire (cas 2-connexe))

Soient $G_\Sigma = (V_\Sigma \cup A_\Sigma, E_\Sigma)$ un hyper-graphe plan 2-connexe et F_Σ l'ensemble des centres des faces de G_Σ .

Le graphe intermédiaire de G est le graphe plan G_I dont l'ensemble des sommets est $V_\Sigma \cup F_\Sigma$ et tel que chaque sommet f de F_Σ soit relié à tous les sommets de V_Σ incidents à la face f .

Par construction, tout chemin compatible avec G_Σ est homotope à un chemin de G_I . En revanche, la propriété « être une courbe de Jordan » n'est pas stable par homotopie ; une courbe de Jordan peut très bien ne pas être homotope à une courbe de Jordan de G_I comme le montre la figure 8.9.

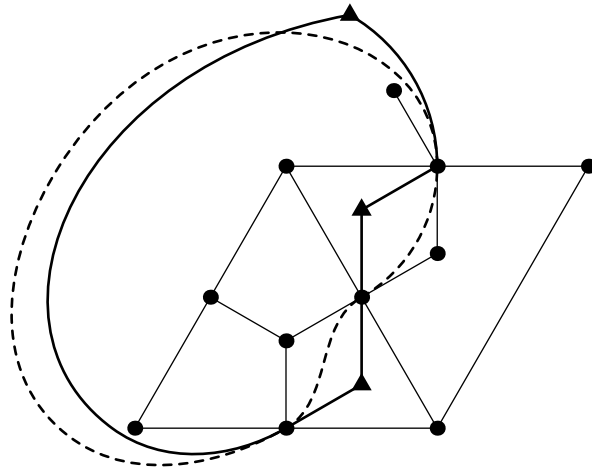
Lemme 8.14 *Un graphe intermédiaire est un graphe plan biparti connexe.*

□ Soit G_I le graphe intermédiaire d'un hyper-graphe plan G_Σ . Le graphe G_I est biparti par construction et sa connexité vient de celle de la surface Σ_{G_Σ} . ■

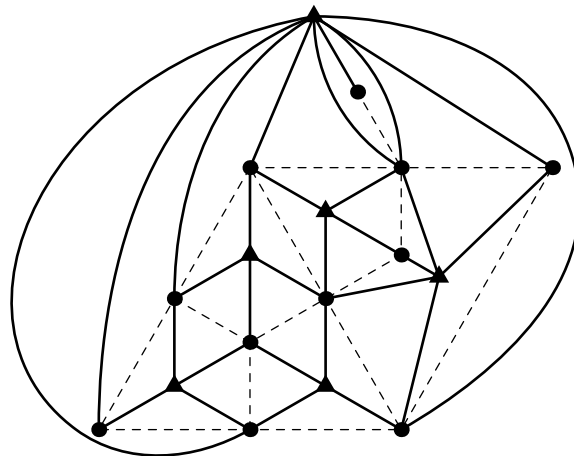
Propriété 8.15 *Soit $G_\Sigma = (V_\Sigma \cup A_\Sigma, E_\Sigma)$ un hyper-graphe plan. Il existe une bijection entre les faces de G_I et les arêtes de G .*

□ Les arêtes de G correspondent aux sommets de A_Σ par définition du graphe d'incidence d'un hyper-graphe. Montrons tout d'abord qu'il y a au plus un sommet de A_Σ par face de G_I .

Soit a dans A_Σ . D'après le lemme 8.9, il existe un lacet μ compatible avec G_Σ qui passe par les sommets de l'hyper-arête a et qui sépare a des sommets de $(A_\Sigma \setminus \{a\}) \cup V_\Sigma$. Ce lacet est homotope à un lacet μ_I de G_I . La composante connexe de $\Sigma \setminus \mu_I$ qui contient a ne

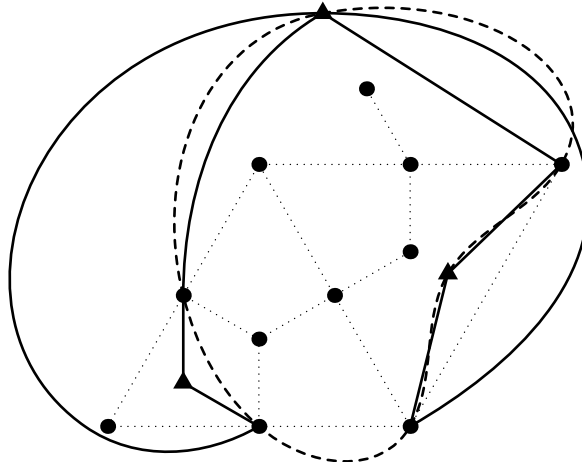


La courbe de Jordan en pointillés est compatible avec le graphe plan. Le lacet en trait plein lui est homotope et passe par les centres des faces.



Nous obtenons le graphe intermédiaire en fixant un représentant par classe d'homotopie de chemins entre le centre d'une face et les sommets qui la bordent. Plusieurs chemins ayant les mêmes extrémités peuvent apparaître.

FIG. 8.8 – Construction du graphe intermédiaire



Le représentant dans le graphe intermédiaire de la courbe de Jordan en pointillés est le lacet singulier en trait plein.

FIG. 8.9 – Une courbe de Jordan homotope à un lacet singulier de G_I

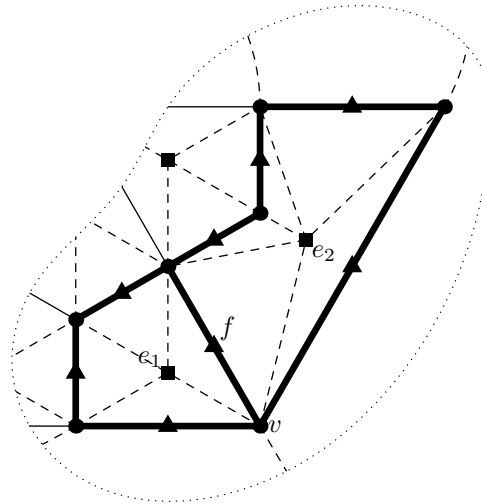
contient aucun autre sommet de A_Σ . Il existe donc une face de G_I qui ne contient que a .

Réciproquement, montrons que toute face de G_I contient au moins un sommet de A_Σ . Supposons qu'il existe une face f de G_I qui ne contienne aucun sommet de A_Σ . Comme toutes les arêtes de G_Σ ont une extrémité dans A_Σ et que le bord de la face a une intersection vide avec A_Σ , il n'existe aucune arête de G_Σ incluse dans cette face.

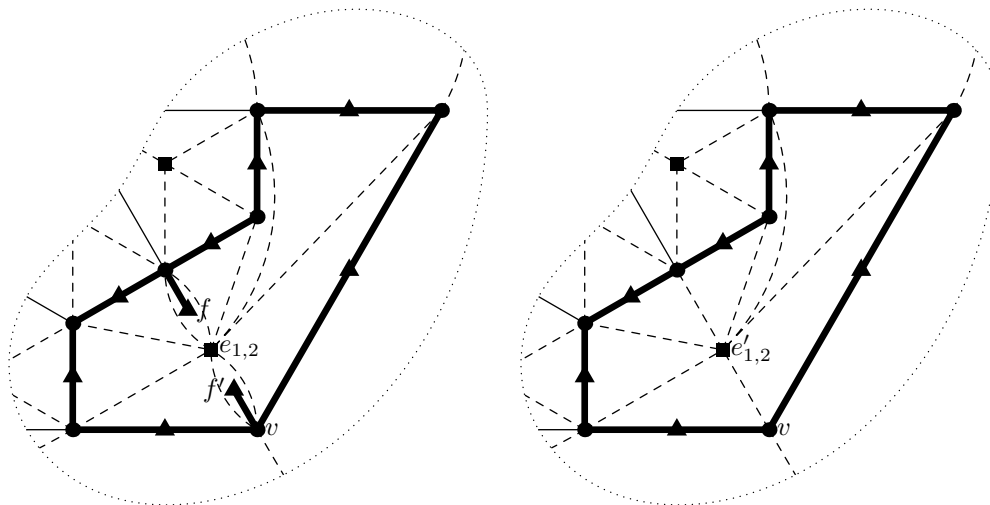
Comme il n'y a aucune arête de G_Σ dans la face f , il existe une face de G_Σ contenant f . Soit x le centre de cette face. Par construction, une des extrémités de chaque arête de G_I est le centre d'une face de G_Σ . Toutes les arêtes de G_I qui constituent le bord de la face f ont donc une extrémité en commun. Comme G_I ne contient pas de boucles, toutes les arêtes du bord de f ont les mêmes extrémités. De plus, comme la face f ne contient pas d'arête de G_Σ , toutes les arêtes du bord de f sont homotopes, ce qui contredit la définition de G_I . ■

Pour retrouver l'arête correspondant à une face f de G_I , il suffit de choisir un représentant par classe d'homotopie entre les chemins reliant le centre de f et les vrais sommets du bord de la face.

Lemme 8.16 (voir fig. 8.10) *Soit G_I un graphe intermédiaire d'un hyper-graphe planaire G et (f, v) une arête de G_I incidente à deux faces correspondant aux hyper-arêtes e_1 et e_2 de G . Le graphe obtenu en remplaçant l'arête (f, v) par l'arête (f', v) où f' est une copie de f est un graphe intermédiaire de l'hyper-graphe G dans lequel nous avons contracté les hyper-arêtes e_1 et e_2 .*



Les arêtes en gras représentent les faces correspondant aux hyper-arêtes e_1 et e_2 .



Dans la figure de gauche, une arête séparant ces faces est « détachée ». La nouvelle face correspond à l'hyper-arête contractée $e_{1,2}$.

Dans celle de droite, deux feuilles du graphe intermédiaire sont supprimées. La face modifiée correspond à l'hyper-arête $e'_{1,2}$.

FIG. 8.10 – Fusion de faces du graphe intermédiaire

□ Soient e l'arête (f, v) , f_1 et f_2 les faces incidentes à (f, v) et a_1 et a_2 les sommets arêtes correspondant à e_1 et e_2 . Nommons f' le point $e(0.5)$ et $e' : t \rightarrow e(.5t)$ le chemin entre v et f' induit par e . Nommons aussi u le point $e(0.75)$.

Comme les faces f_1 et f_2 sont connexes par arcs, il existe un chemin de a_1 à a_2 compatible avec le graphe G_I auquel nous avons rajouté le sommet f' et dans lequel nous avons remplacé l'arête e par l'arête e' . En contractant ce chemin, nous obtenons une représentation plane de l'hyper-graphe planaire G dans lequel nous avons contracté e_1 et e_2 en une hyper-arête $e_{1,2}$. ■

Lemme 8.17 (voir fig. 8.10) *Soit G_I un graphe intermédiaire d'un hyper-graphe planaire G et e une hyper-arête de G correspondant à une face de G_I possédant au moins une feuille v . Le graphe G_I dans lequel nous avons retiré la feuille v est un graphe intermédiaire de l'hyper-graphe G dans lequel l'hyper-arête e est remplacée par une hyper-arête ayant une extrémité de moins.*

□ Soient G_Σ l'hyper-graphe plan correspondant à G_I et a_e le sommet arête de G_Σ correspondant à l'hyper-arête e .

- Si la feuille retirée est un sommet face u de G_I , alors deux chemins de G_Σ entre a_e et u deviennent homotopes. Il suffit de retirer l'un des deux, ce qui revient à retirer une des extrémités de l'hyper-arête e dans G , pour obtenir un hyper-graphe plan correspondant au graphe intermédiaire G_I amputé;
- dans le cas contraire, la feuille est un sommet initial u de G_I et il suffit de retirer de G_Σ l'arête entre a_e et u pour obtenir un hyper-graphe plan correspondant au graphe intermédiaire G_I amputé.

Dans les deux cas, le graphe obtenu est bien un graphe intermédiaire de G dans lequel l'hyper-arête e a été amputée d'une de ses extrémités. ■

De ces deux lemmes, nous tirons le corollaire suivant :

Corollaire 8.18 *Soient G_I un graphe intermédiaire d'un hyper-graphe planaire G et μ une courbe de Jordan de G_I induisant une séparation (E_1, E_2) de G . Le graphe G_I dans lequel toutes les arêtes et tous les sommets contenus dans le domaine correspondant aux arêtes de E_2 sont supprimés est un graphe intermédiaire de l'hyper-graphe G dans lequel nous avons contracté les arêtes de E_2 en une seule hyper-arête contenant la frontière $\partial(E_1, E_2)$.*

8.1.4 Séparateurs minimaux des graphes planaires

Les courbes de Jordan compatibles avec un hyper-graphe plan permettent de représenter certaines séparations de cet hyper-graphe. Nous montrons ici que c'est en particulier le cas pour les séparateurs minimaux.

Comme les cycles élémentaires du graphe intermédiaire représentent en général des séparations, nous donnons une caractérisation des cycles élémentaires du graphe intermédiaire correspondant à des séparateurs minimaux.

Propriété et définition 8.19 (Réalisation d'un séparateur minimal)

Soient $G_\Sigma = (V_\Sigma \cup A_\Sigma, E_\Sigma)$ un hyper-graphe plan et S un a, b -séparateur minimal de G . Il existe une courbe de Jordan μ compatible avec G_Σ – une réalisation de S – telle que :

- (i) l'ensemble $V_{G_\Sigma}(\mu)$ soit égal à S ;
- (ii) les sommets a et b ne soient pas dans la même composante connexe de $\Sigma \setminus \mu$.

□ Soit $C^a(S)$ la composante connexe de $G \setminus S$ contenant a et $\tilde{C}_\Sigma^a(S)$ l'ensemble des arêtes de G_Σ incidentes aux sommets de $C^a(S)$. L'hyper-graphe G_Σ^a obtenu à partir de G_Σ en contractant les arêtes de $\tilde{C}_\Sigma^a(S)$ est plan. Soit \tilde{a} le sommet résultant de cette contraction. D'après le lemme 8.9, il existe un lacet μ^a compatible avec G_Σ^a qui sépare \tilde{a} des autres sommets de G_Σ^a . En « décontractant » les arêtes de $\tilde{C}_\Sigma^a(S)$, nous obtenons à partir de μ^a un lacet μ compatible avec G_Σ qui sépare $C^a(S)$ et l'ensemble des sommets arêtes voisins de $C^a(S)$ du reste des sommets de G_Σ .

Soit Σ^a la région de $\Sigma \setminus \mu$ contenant $C^a(S)$ et Σ^b celle qui contient $C^b(S)$. Notons que la frontière de Σ^a est μ . La frontière μ' de Σ^b est incluse dans celle de Σ^a . Soit Σ'^a la région de $\Sigma \setminus \mu'$ contenant Σ^a . Comme Σ'^a et Σ^b sont deux ouverts connexes par arcs disjoints ayant la même frontière, ils sont homéomorphes à des disques ouverts et leur frontière est une courbe de Jordan. Par construction, celle-ci est compatible avec G_Σ . Comme μ' est la frontière de Σ^b , l'ensemble $V(\mu')$ est égal à S et μ' sépare a et b . ■

Grâce au graphe intermédiaire, nous pouvons restreindre le nombre de courbes de Jordan pouvant réaliser un séparateur minimal. Cependant, cette restriction n'est pas suffisante : un même séparateur minimal peut admettre plusieurs réalisations comme le montre la figure 8.11. Ceci nous amène à choisir certaines réalisations particulières pour un séparateur minimal donné.

Propriété et définition 8.20 (Réalisation proche d'un sommet)

Soient G_I un graphe intermédiaire d'un hyper-graphe planaire $G = (V, E)$ et S un a, b -séparateur minimal de G . Toute réalisation de S est homotope à une réalisation μ_I de S dans G_I . De plus, il existe une unique réalisation μ_I^a de S dans G_I – la réalisation de S proche de a – telle que si μ_I' est une autre réalisation de S dans G_I , alors μ_I^a n'intersecte que la composante connexe de $\Sigma \setminus \mu_I'$ contenant a .

□ Soient μ une réalisation de S et μ_I son représentant dans G_I . Supposons par l'absurde que μ_I ne soit pas une courbe de Jordan. La frontière μ_I' d'une des deux composantes connexes de $\Sigma \setminus \mu_I$ contenant a et b n'est pas égale à

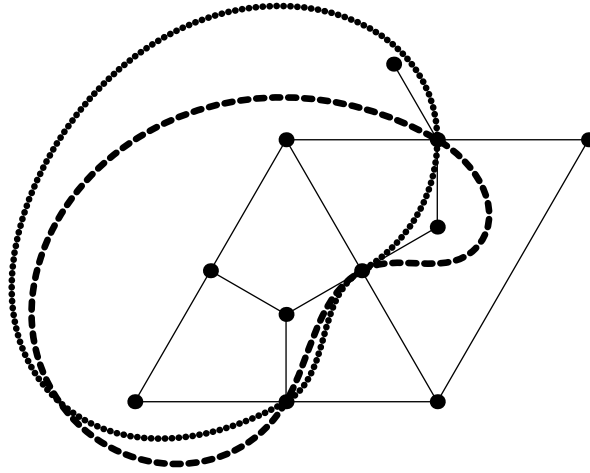


FIG. 8.11 – Deux réalisations non homotopes d'un même séparateur minimal

μ_I . Par conséquent, comme G_I est 2-connexe, l'ensemble $V(\mu'_I)$ sépare a et b et est strictement inclus dans $V(\mu_I)$ ce qui contredit le fait que S soit un a, b -séparateur minimal.

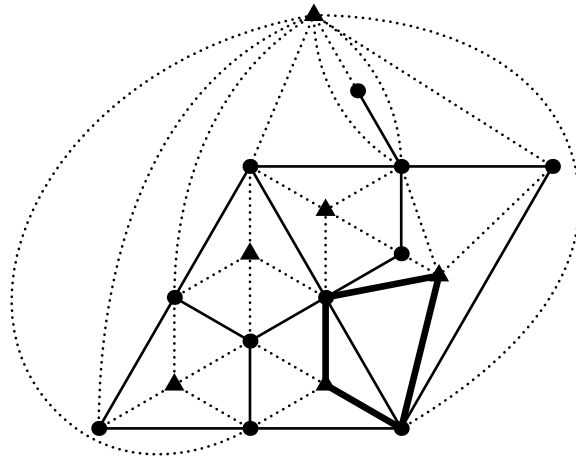
Il y a un nombre fini de réalisations de S dans μ_I . Notons Σ_μ^a la composante connexe de $\Sigma \setminus \mu$ contenant a . Soient μ_1 et μ_2 deux réalisations de S dans G_I et posons que μ_1 est plus proche de a que μ_2 si $\Sigma_{\mu_1}^a$ est inclus dans $\Sigma_{\mu_2}^a$. Comme μ_2 est une courbe de Jordan, $\mu_2 \cap \Sigma_{\mu_1}^a$ ne contient pas de boucles. Par conséquent, le voisinage μ_3 de la composante connexe de $\Sigma \setminus (\mu_1 \cup \mu_2)$ contenant a est une courbe de Jordan et c'est une réalisation de S plus proche de a que μ_1 et μ_2 . Ceci implique qu'il en existe une plus proche de a que toute les autres, ce qui achève la démonstration. ■

La propriété 8.20 montre que tous les séparateurs minimaux de G sont réalisés par des courbes de Jordan de G_I . Cependant, la réciproque n'est pas vraie. Nous donnons deux conditions permettant de rejeter certains cycles de G_I ne réalisant pas de séparateurs minimaux.

Définition 8.21 (Cycle séparant)

Soit G_I un graphe intermédiaire d'un hyper-graphe planaire. Un cycle élémentaire $(v_1, f_1 \dots, v_p, f_p)$ de G_I est séparant si v_1 a un voisin dans chacune des composantes connexes de Σ privé du cycle.

Par définition (cf. 8.19), une réalisation d'un séparateur minimal S est un cycle séparant. Le plus souvent, un cycle séparant ne passe pas par les sommets d'un séparateur minimal. Cependant, dans le cas d'un séparateur de taille deux, cela peut arriver (il suffit que le séparateur soit une arête et que le cycle « fasse le tour » de l'arête).



Le cycle en gras est un cycle élémentaire du graphe intermédiaire qui ne sépare pas de sommets.

FIG. 8.12 – Un cycle non-séparant

Lemme 8.22 Soient G_I un graphe intermédiaire d'un hyper-graphe plan G et S un séparateur minimal de G . Si μ n'est pas un cycle séparant, alors μ ne réalise pas S .

Définition 8.23 (court-circuit)

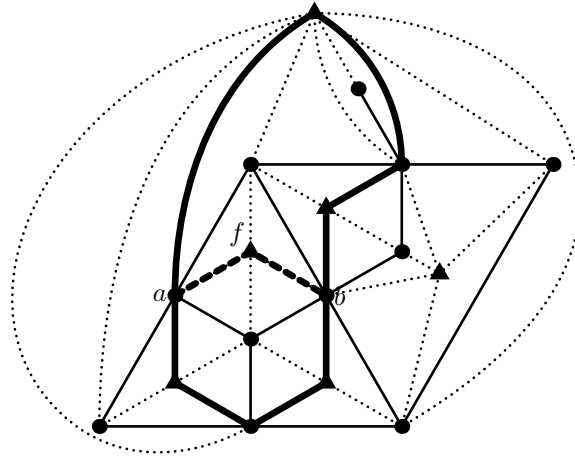
Soient G_I un graphe intermédiaire d'un hyper-graphe planaire et μ un cycle élémentaire de G_I . Un court-circuit de μ est le centre f d'une face incidente à deux sommets de μ qui ne sont pas à distance deux sur μ .

Lemme 8.24 Soit G_I un graphe intermédiaire d'un hyper-graphe planaire G . Si f est un court-circuit d'un cycle élémentaire μ , alors $V(\mu)$ n'est pas un séparateur minimal de G .

□ Notons $(v_1, f_2, v_2, \dots, v_p, f_p)$ le cycle μ et supposons par l'absurde que $V(\mu)$ soit un a, b -séparateur minimal de G . Nous pouvons supposer que le court-circuit f est adjacent à v_1 et v_i ($i > 2$). Nous pouvons donc définir deux cycles μ_1 et μ_2 égaux respectivement à $(v_1, f_1, \dots, v_i, f)$ et $(v_1, f, v_i, f_i, \dots, f_p)$. L'un de ces deux cycles μ_i sépare a et b . L'ensemble $V(\mu_i)$ est donc un a, b -séparateur. Mais cet ensemble est strictement inclus dans $V(\mu)$ ce qui contredit le fait que $V(\mu)$ soit un a, b -séparateur minimal. ■

Définition 8.25 (Séparateur ordonné)

Soit G_Σ un hyper-graphe plan. Un séparateur ordonné de G_Σ est un cycle séparant μ de G_I sans court-circuit. Il réalise l'ensemble $V_{G_\Sigma}(\mu)$.



Le cycle en gras est un cycle élémentaire du graphe intermédiaire.

Le sommet f est adjacent à a et b . C'est un court-circuit.

FIG. 8.13 – Un court-circuit

Propriété 8.26 Soient G_I un graphe intermédiaire d'un hyper-graphe planaire G et μ_I une courbe de Jordan de G_I . Un ensemble S est un séparateur minimal si et seulement si il est réalisé par un séparateur ordonné.

□ Si S est un séparateur minimal, d'après la propriété 8.20, il existe une courbe de Jordan μ_I de G_I qui réalise S . Les lemmes 8.22 et 8.24 montrent que μ_I est un séparateur ordonné.

Réciproquement, supposons que S soit la réalisation d'un cycle séparant de G_I sans court-circuit μ_I . Notons $(v_1, f_1, \dots, v_p, f_p)$ ce cycle. Comme μ_I est un cycle séparant, l'ensemble S est un séparateur de G . Montrons par induction sur le nombre k de composantes connexes de $G \setminus S$ que S est un séparateur minimal.

Si $k = 2$, comme μ_I est un cycle séparant, soient a et b deux sommets voisins de v_0 dans chacune des deux composantes de $\Sigma_G \setminus \mu_I$. Il existe un a, b -séparateur minimal S' inclus dans S . Celui-ci est réalisé par une courbe de Jordan ν_I de G_I . Si S' est strictement inclus dans S , il existe un indice i tel que v_i appartienne à S' mais pas v_{i+1} . Sur ν_I l'une des deux faces incidentes à v_i est un court-circuit de μ_I , ce qui est absurde. Par conséquent, S est un séparateur minimal.

Si $k > 2$, soit S' un séparateur minimal inclus dans S . Si S' est égal à S , la propriété est vraie. Sinon, d'après la propriété 8.5, S' est de taille au plus deux. Il existe une composante connexe C de $G \setminus S'$ dont μ_I ne rencontre aucun sommet. Le cycle μ_I est un cycle séparant sans court-circuit de $G \setminus C$. Par induction, S est un séparateur minimal. ■

8.2 Énumération des séparateurs minimaux

Nous disposons maintenant des outils nous permettant d'adapter l'algorithme 3.2 que nous avons vu au chapitre 3.

Nous montrons que dans le cas des hyper-graphes planaires 3-connexes, il est possible de spécialiser l'algorithme 3.3 de façon à ce qu'il fonctionne en $O(n)$ par séparateur. De plus, d'après la propriété 8.5, les séparateurs minimaux d'un tel hyper-graphe sont minimaux pour l'inclusion. Par conséquent, si S est un séparateur minimal d'un hyper-graphe planaire 3-connexe G , soit S est un $a, *$ -séparateur minimal, soit a appartient à S . Dans ce second cas, d'après la propriété 2.10, S est un $b, *$ -séparateur minimal pour n'importe quel b dans une composante pleine de S . Nous pouvons donc choisir b dans le voisinage de a . Nous avons donc :

$$\Delta_G = \mathcal{S}_a \bigcup_{b \in N(a)} \mathcal{S}_b$$

et en choisissant a de degré cinq ou moins, nous obtiendrons un séparateur minimal au plus cinq fois. Nous pouvons donc énumérer les séparateurs minimaux des hyper-graphes planaires 3-connexes en temps $O(n)$ par séparateur ce qui est mieux que l'algorithme [BBC00]. L'algorithme que nous présentons ici est spécifique aux hyper-graphes planaires 3-connexes mais il est possible de le généraliser à tous les hyper-graphes planaires.

Algorithme 8.1 `calc_a*_aux` : Énumération de $\mathcal{S}_{a, B_S}(S, O)$

entrée :

$G = (V, E)$ un hyper-graphe
 $a \in V$
 S un $a, *$ -séparateur minimal
 $O \subseteq S$

sortie :

$\mathcal{S}_{a, B_S}(S, O)$

début

$\mathcal{S} \leftarrow \emptyset$

pour chaque $x \in S \setminus O$ **faire**

si $\mathcal{S}_{a, B_S}^x(S, O) \neq \emptyset$ **alors**

pour chaque S_i élément minimal de $\mathcal{S}_{a, B_S}^x(S, O)$ **faire**

$\mathcal{S} \leftarrow \mathcal{S} \cup \text{calc_a*_aux}(G, a, S_i, O)$

$O \leftarrow O \cup \{x\}$

rendre $\mathcal{S} \cup \{S\}$

fin

Jusqu'à présent, nous choisissons le sommet que nous allons pousser au hasard parmi les sommets de $S \setminus O$. Les sommets de l'ensemble O étaient donc distribués de façon aléatoire dans S . Comme un séparateur minimal S d'un hyper-graphe planaire est réalisé par un séparateur ordonné μ_I de G_I , nous pouvons choisir de pousser les sommets de S en suivant l'ordre induit par μ_I . De cette façon, l'ensemble O induit « une portion continue » de μ_I . Nous montrons que cette stratégie nous permet de tester si l'ensemble $S_{a, B_S}^x(S, O)$ est vide en temps $O(d(x))$.

Comme nous fixons un sommet a , nous pouvons orienter les réalisations des $a, *$ -séparateurs minimaux dans le sens trigonométrique autour de a .

Définitions 8.27 (Intervalle d'un séparateur)

Soient S un séparateur minimal d'un hyper-graphe planaire et une réalisation $(v_1, f_1, \dots, v_p, f_p)$ de S dans le sens trigonométrique. Un intervalle de S est un ensemble de sommets O correspondant à une suite $(v_i, v_{i+1}, \dots, v_j)$ de sommets de S . Le début et la fin de O sont respectivement les sommets v_i et v_j . L'intérieur de O est l'ensemble des sommets de O privé de v_i et v_j . Le sommet suivant O est le sommet v_{j+1} .

La notion d'intervalle d'un séparateur S semble dépendre du choix de la réalisation de S mais ce n'est pas le cas.

Propriété 8.28 Soient G_I un graphe intermédiaire d'un hyper-graphe planaire G , S un a, b -séparateur minimal de G , O un intervalle de S et x le sommet suivant O . S'il existe un sommet de $N(x) \setminus C_G^a(S)$ incident à la même face qu'un sommet de l'intérieur de O , l'ensemble $S_{a, B_S}^x(S, O)$ est vide.

□ Supposons par l'absurde qu'un sommet y de $N(x) \setminus C_G^a(S)$ et un sommet v de l'intérieur de O soient incidents à une même face f et qu'il existe un a, b -séparateur minimal S' dans $S_{a, B_S}^x(S, O)$.

Le séparateur minimal S est réalisé par un séparateur ordonné μ_S de G_I . Notons Σ^b la composante connexe de $G \setminus \mu_S$ contenant b . Soit ν un chemin de Σ^b compatible avec G_Σ passant par les sommets x, y, v ainsi que par la face f . Ce chemin délimite deux domaines Σ_1^b et Σ_2^b de Σ^b . La réalisation $\mu_{S'}$ de S' proche de b est incluse dans Σ^b . De plus, comme elle passe par les sommets de O , elle intersecte à la fois Σ_1^b et Σ_2^b . Comme c'est une courbe de Jordan et qu'elle ne passe pas par x , elle passe à la fois par y et par v . Mais alors f est un court-circuit de $\mu_{S'}$ ce qui contredit le fait que S' soit un séparateur minimal. ■

Théorème 8.29

Soient G_I un graphe intermédiaire d'un hyper-graphe planaire G , S un a, b -séparateur minimal de G , O un intervalle de S et x le sommet suivant O . L'ensemble $S_{a, B_S}^x(S, O)$ est non vide si et seulement si :

Algorithme 8.2 `calc_a*_planaire_aux` : Énumération de $\mathcal{S}_{a,B_S}(S, O)$

entrée :

$G = (V, E)$ un hyper-graphe

$a \in V$

$S = (v_1, \dots, v_p)$ un $a, *$ -séparateur minimal ordonné

$O = (v_1, \dots, v_i)$ dans le sens trigonométrique

Les sommets de G sont étiquetés par le plus petit $i > 1$ tel qu'ils soient incidents à une même face que v_i .

Les sommets de $C_G^a(S)$ sont aussi marqués.

sortie :

$\mathcal{S}_{a,B_S}(S, O)$

début

pour j de $i + 1$ à p

$\mathcal{S} \leftarrow \text{calc_a*_planaire_aux}(G, a, S, (v_1, \dots, v_j))$

si $\forall y \in N(v_j) \setminus C_G^a(S)$, y n'est pas étiqueté $1 < k < i$ **alors**

pour chaque S_i élément minimal de $\mathcal{S}_{a,B_S}^{v_j}(S, O)$ **faire**

calculer les étiquettes de G correspondant à S_i

$\mathcal{S} \leftarrow \mathcal{S} \cup \text{calc_a*_planaire_aux}(G, a, S_i, O)$

rendre \mathcal{S}

fin

- il n’y a pas de face incidente à la fois à un sommet de $N(x) \setminus C_G^a(S)$ et à un sommet de l’intérieur de O ;
- il existe un voisin du début de O dans B_S qui n’est pas voisin de x .

□ Supposons que l’ensemble $S_{a,B_S}^x(S, O)$ ne soit pas vide. Soit S' un de ses éléments. La propriété 8.28 montre que la première condition est remplie. De plus, l’ensemble O est inclus dans S' donc v appartient à S' . Il existe un voisin de v dans B_S qui n’est pas voisin de x .

Réciproquement, supposons que les deux conditions soient remplies. Soit $(v_1, f_1, \dots, v_p, f_p)$ une réalisation de S dans G_I notée μ_I telle que O soit l’ensemble $\{v_1, \dots, v_i\}$, v soit égal à v_1 et x soit égal à v_{i+1} .

Numérotons les voisins y_1, \dots, y_l de x dans B_S dans le sens trigonométrique. Les sommets y_1 et v_i sont incidents à une même face, c’est aussi le cas des sommets y_l et v_{i+2} et des sommets y_j et y_{j+1} . Si nous notons g_0 une face incidente à v_i et y_1 , g_i une face incidente à y_i et à y_{i+1} et g_l une face incidente à y_l et v_{i+2} , $(v_0, f_0, \dots, v_i, g_0, y_1, g_1, \dots, y_l, g_l, v_{i+2}, \dots, f_p)$ est un cycle de G_I . Soit μ_I un tel cycle passant par les sommets de $S \setminus \{x\}$ et ceux de $N(x) \cap B_S$ et contenant le début de O de longueur minimale. D’après la condition 8.29, le cycle μ_I passe par tous les sommets de O . De plus il n’a pas de court-circuit sans quoi il ne serait pas de longueur minimale. D’après la propriété 8.26, la seule condition pour que μ_I réalise un séparateur minimal est que μ_I soit un cycle séparant. Or la condition 8.29 énonce justement ce fait. La réalisation S' de μ_I appartient à $S_{a,B_S}^x(S, O)$ ce qui achève la démonstration du résultat. ■

Le théorème 8.29 fournit un moyen efficace pour décider si l’ensemble $S_{a,B_S}^x(S, O)$ est vide ou non. Pour cela, il faut pouvoir trouver les sommets qui peuvent servir à créer un court-circuit. Comme ces sommets ne dépendent pas du sommet que l’on pousse mais uniquement des sommets de l’ensemble O , nous pouvons les précalculer et les étiqueter. C’est exactement ce que fait l’algorithme 8.2. La correction de cet algorithme découle donc directement du théorème 8.29.

Théorème 8.30

L’algorithme 8.2 est correct et fonctionne en temps $O(n)$ par séparateur.

- Les calculs effectués pour traiter un séparateur minimal sont :
- l’obtention du séparateur minimal S_i lorsque l’on calcule les éléments minimaux d’un ensemble $S_{a,B_S}^{v_j}(S, O)$: cette étape se fait à l’aide d’un parcours de G et coûte $O(n + m)$;
 - le calcul des étiquettes des sommets de G suivant S_i : cette étape peut aussi se faire en un parcours ;
 - le parcours dans la boucle **pour** : chaque passage coûte $d(v_j)$. Au final, cette étape coûte au plus $\sum_{v \in V} d(v)$ qui est égal à m .

Le coût de traitement d'un séparateur minimal est donc de $O(n + m)$ mais comme dans un hyper-graphe planaire, m vaut au plus $3n - 6$, l'algorithme 8.2 fonctionne en temps $O(n)$ par séparateur. ■

8.3 Représentations de décompositions

Dans cette partie, nous considérons des hyper-graphes fortement arête-connexes.

Définition 8.31 (Hyper-graphe fortement arête-connexe)

Un hyper-graphe G est fortement arête-connexe s'il reste connexe même privé des extrémités d'une hyper-arête et si aucune hyper-arête n'en contient d'autre.

Une courbe de Jordan de G_I définit une séparation de G . Pour obtenir une matriochka, se donner une famille \mathcal{F} de courbes de Jordan de G_I n'est pas suffisant. Il faut que les séparations induites par \mathcal{F} ne se croisent pas.

Définition 8.32 (Courbes de Jordan parallèles)

Deux courbes de Jordan sont parallèles si et seulement si chacune n'intersecte qu'un des deux domaines que définit l'autre.

Définition 8.33 (Θ -structure)

Une Θ -structure d'un graphe intermédiaire G_I est la donnée de trois chemins ν_1 , ν_2 et ν_3 d'intérieurs disjoints reliant deux sommets distincts x et y .

Une Θ -structure $\{\nu_1, \nu_2, \nu_3\}$ induit trois courbes de Jordan de G_I : $\nu_2 \cdot (\nu_3^{-1})$, $\nu_1 \cdot (\nu_3^{-1})$ et $\nu_1 \cdot (\nu_2^{-1})$.

Le nom de Θ -structure provient du fait que les trois chemins ν_1 , ν_2 et ν_3 dessinent un Θ .

Propriété 8.34 Une famille \mathcal{F} de courbes de Jordan deux à deux parallèles de G_I définit une matriochka de G .

□ Comme nous l'avons déjà mentionné, chaque courbe μ de la famille \mathcal{F} définit une séparation (E_μ^1, E_μ^2) de G . La famille \mathcal{F} définit donc une famille \mathcal{C} de sous-ensembles de E telle que si X est un élément de \mathcal{C} , $E \setminus X$ l'est aussi. De plus, considérons X et Y deux éléments de \mathcal{C} . Les ensembles X et Y correspondent à deux domaines induits par deux courbes μ_X et μ_Y de \mathcal{F} . Quitte à considérer les complémentaires de X et de Y , nous pouvons supposer que X et Y correspondent respectivement aux domaines induits par μ_X et μ_Y ne contenant pas μ_Y et μ_X . Ainsi, X et Y sont disjoints et ne se chevauchent pas. Les ensembles X et Y étant quelconques, la famille \mathcal{C} est une matriochka de G . ■

Nous montrons ici qu'une famille de courbes de Jordan de G_I induit une matriochka. Plus précisément, une famille de courbes de Jordan de G_I maximale pour l'inclusion induit une matriochka complète de G .

Propriété 8.35 *Soient \mathcal{F} une famille de courbes de Jordan de G_I deux à deux parallèles maximale pour l'inclusion, μ une courbe de \mathcal{F} et Σ_μ l'un des deux domaines que délimite μ .*

- soit Σ_μ ne contient qu'une face de G_I ;
- soit il existe une Θ -structure de G_I incluse dans Σ_μ qui induit trois courbes de Jordan de \mathcal{F} dont la courbe μ .

□ Considérons l'ensemble \mathcal{F}_{Σ_μ} des courbes de Jordan de \mathcal{F} délimitant un domaine inclus dans Σ_μ et supposons que Σ_μ contienne au moins deux faces de G_I .

Il existe une face f de G_I incluse dans Σ_μ incidente à une arête de μ . Comme d'après la propriété 8.15, la face f correspond à une arête de G et que G est fortement arête-connexe, les sommets de $V_\Sigma \cap \mu$ ne sont pas tous incidents à f . Il existe donc une courbe μ' de \mathcal{F}_μ distincte de μ qui définit un domaine le plus grand possible contenant f . La courbe μ' partage avec μ au moins une arête et comme μ' est distincte de μ , il existe deux sommets x et y de μ et un chemin ν_1 de μ' de x et y dont aucune arête n'appartient à μ . Les deux sommets x et y découpent le cycle μ en deux chemins ν_2 et ν_3 de x à y .

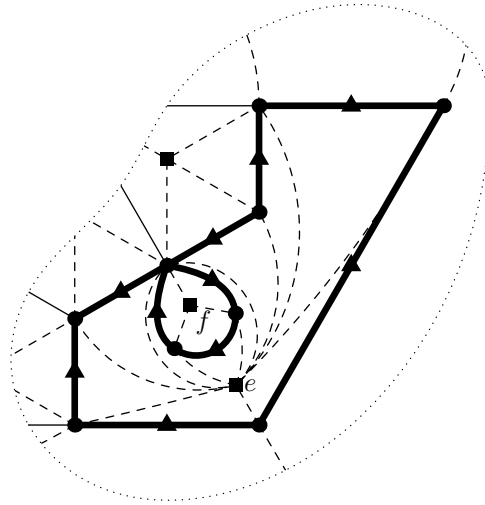
Ces trois chemins définissent trois courbes de Jordan de G_I : μ , μ_1 et μ_2 égales respectivement à $\nu_2 \cdot (\nu_3^{-1})$, $\nu_1 \cdot (\nu_2^{-1})$ et $\nu_1 \cdot (\nu_3^{-1})$. L'un des domaines de Σ_μ défini par μ_1 et μ_2 contient celui défini par μ' . Supposons que ce soit celui correspondant à μ_1 . Si un lacet croise μ_1 mais pas μ et μ' , il contient un point singulier et n'est donc pas une courbe de \mathcal{F} . Comme aucune courbe de \mathcal{F} ne croise μ_1 et que \mathcal{F} est maximal pour l'inclusion, μ_1 est un élément de \mathcal{F} . Le même raisonnement montre que c'est aussi le cas pour μ_2 , ce qui achève la démonstration. ■

Si nous ne supposons pas l'hyper-graphe fortement arête-connexe, la propriété n'est pas vraie comme l'illustre la figure 8.14.

Nous déduisons des propriétés 8.34 et 8.35 le corollaire suivant :

Corollaire 8.36 *Une famille de courbes de Jordan de G_I deux à deux parallèles maximale pour l'inclusion définit une décomposition en branches.*

Si \mathcal{F} est une telle famille maximale pour l'inclusion, nous en déduisons une décomposition en branches T_τ de G . Les cliques maximales de la triangulation $H_\mathcal{F}$ induite par T_τ correspondent aux bords des faces de G_I et aux Θ -structures induites par \mathcal{F} .



L'hyper-arête f est incluse dans l'hyper-arête e . Ceci crée une « boucle » dans la face de G_I correspondant à e . La situation serait la même si f était le résultat d'une contraction de sommets auquel cas l'hyper-arête e déconnecterait l'hyper-graphe.

FIG. 8.14 – Un graphe intermédiaire d'un hyper-graphe non fortement arête-connexe.

Maintenant que nous savons exprimer certaines décompositions en branches à partir de courbes de Jordan du graphe intermédiaire, il est naturel de se demander :

- s'il existe une telle décomposition qui soit de profil minimal ou tout du moins de largeur de branches minimale ;
- s'il existe une telle décomposition qui induise une décomposition arborescente de largeur arborescente minimale.

Seymour et Thomas [ST94] ont répondu en partie à la première question en montrant qu'il existe une « bond-decomposition » de largeur de branches minimale. Nous allons répondre par l'affirmative à la seconde question.

Théorème 8.37 ([ST94])

Soit G_Σ un hyper-graphe plan fortement arête-connexe. Il existe une décomposition en branches de G_Σ de largeur minimale induite par une famille de courbes de Jordan de G_I deux à deux parallèles.

Théorème 8.38

Soient G_Σ une représentation plane d'un hyper-graphe planaire G et H une triangulation minimale de G . Il existe une famille \mathcal{F} de courbes de Jordan de G_I telle que la triangulation $H_{\mathcal{F}}$ soit égale à H .

□ D'après le théorème 5.3 de Parra et Scheffler, H est donné par la famille Δ_H de ses séparateurs minimaux. Montrons le résultat par induction sur le nombre k de composantes pleines qu'induisent les séparateurs minimaux de Δ_H .

Si k est égal à 0, la famille Δ_H est vide et l'hyper-graphe G est triangulé. La famille vide \mathcal{F} convient. Sinon, il existe un a, b -séparateur minimal S dans Δ_H . D'après les propriétés 8.19 et 8.20, il existe une courbe de Jordan μ_S qui réalise S ; cette courbe de Jordan induit une séparation (E_1, E_2) de G . Notons G_1 et G_2 l'hyper-graphe G dans lequel nous avons contracté respectivement les ensembles E_2 et E_1 en l'hyper-arête S . Soient \mathcal{S}_1 et \mathcal{S}_2 les séparateurs minimaux de Δ_H qui sont des séparateurs minimaux respectivement de G_1 et de G_2 . La famille Δ_H étant maximale pour l'inclusion, c'est aussi le cas des familles \mathcal{S}_1 et \mathcal{S}_2 ; ces deux familles induisent donc des triangulations minimales H_1 et H_2 de G_1 et G_2 . De plus, si nous notons $(V_1, E_1 \cup \{S\})$ et $(V_2, E_2 \cup \{S\})$ les hyper-graphes H_1 et H_2 , l'hyper-graphe $(V_1 \cup V_2, E_1 \cup E_2)_{\{S\}}$ est exactement la triangulation H .

D'après le corollaire 8.18, nous pouvons obtenir un graphe intermédiaire G_{1I} de G_1 et G_{2I} de G_2 en retirant à G_I les sommets et les arêtes incluses dans le domaine de μ_S correspondant respectivement aux hyper-arêtes de E_2 et E_1 . De plus, les familles \mathcal{S}_1 et \mathcal{S}_2 induisent moins de composantes pleines que Δ_H . Par hypothèse, il existe des familles \mathcal{F}_1 et \mathcal{F}_2 de courbes de Jordan de G_{1I} et G_{2I} telles que $H_{\mathcal{F}_1}$ et $H_{\mathcal{F}_2}$ soient exactement les triangulations H_1 et H_2 . Soit \mathcal{F} la famille $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \{S\}$. Par construction, μ est parallèle à toutes les courbes de \mathcal{F}_1 et de \mathcal{F}_2 . Les courbes de Jordan de \mathcal{F} sont donc toutes parallèles deux à deux. De plus, H est égale à la triangulation $H_{\mathcal{F}}$. ■

Le théorème 8.38 montre que toute triangulation minimale H de G est induite par une matriochka obtenue à partir d'une famille \mathcal{F} de courbes de Jordan. Le théorème 4.23 montre que si nous complétons \mathcal{F} en \mathcal{F}' , la triangulation $H_{\mathcal{F}'}$ est une sous-triangulation de $H_{\mathcal{F}}$. Le corollaire 8.36 permet donc d'affirmer :

Théorème 8.39

Soient H une triangulation minimale d'un hyper-graphe planaire fortement arête-connexe G et G_I un graphe intermédiaire de G . Il existe une décomposition en branches induite par une famille de courbes de Jordan de G_I telle que la triangulation $H_{\mathcal{T}}$ soit exactement H .

8.4 Dualité et largeur arborescente

8.4.1 Définition et théorème de dualité

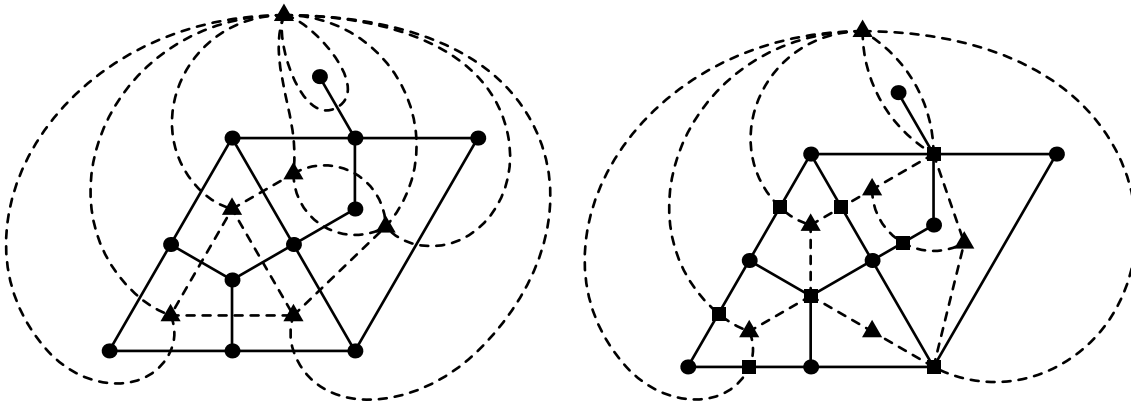
La dualité pour les graphes planaires est une notion classique (voir par exemple [Die00]). Les sommets du graphe dual G^* d'un graphe planaire G

correspondent aux faces de G et à chaque arête de G qui borde les faces f_1 et f_2 est associée l'arête duale (f_1, f_2) . Notons que l'arête duale d'un isthme – c'est-à-dire une arête qui déconnecte le graphe – est une boucle. Dans le cas des hyper-graphes, nous pourrions choisir de représenter les boucles par une hyper-arête à un seul sommet mais alors une hyper-arête e et son hyper-arête duale e^* n'auraient pas nécessairement le même nombre d'extrémités et retrouver e à partir de e^* deviendrait impossible. Il est donc important que les hyper-arêtes d'un hyper-graphe planaire soient des multi-ensembles de sommets. La figure 8.15 présente un graphe et un hyper-graphe plan ainsi que leurs duaux respectifs.

Définition 8.40 (Hyper-graphe dual)

Soient $G_\Sigma = (V_\Sigma \cup A_\Sigma, E_\Sigma)$ un hyper-graphe plan et F_Σ l'ensemble des centres des faces de G_Σ .

Pour chaque face F de G_Σ dont le centre est f et chaque sommet arête a de A_Σ incident à F , il est possible de choisir un représentant par classe d'homotopie de chemins de a à f dans F de telle sorte que l'hyper-graphe $G_\Sigma^* = (F_\Sigma \cup A_\Sigma, E_\Sigma^*)$ dont les arêtes sont ces représentants soit plan. Cet hyper-graphe plan est l'hyper-graphe dual de G_Σ .



La figure de gauche présente un graphe plan et son dual.
Celle de droite présente un hyper-graphe plan ainsi que son dual.

FIG. 8.15 – Graphes et hyper-graphes duaux

Théorème 8.41

Soit $G_\Sigma = (V_\Sigma \cup A_\Sigma, E_\Sigma)$ un hyper-graphe plan.

$$(G_\Sigma^*)^* = G_\Sigma.$$

Avant de démontrer ce théorème, nous avons besoin de la propriété 8.42 et du lemme 8.43 suivant.

Propriété 8.42 Soit $G_\Sigma = (V_\Sigma \cup A_\Sigma, E_\Sigma)$ un hyper-graphe plan. Il y a une bijection entre les faces de G_Σ^* et les sommets de V_Σ .

□ Montrons tout d'abord qu'il y a au plus un sommet de V_Σ par face de G_Σ^* . Pour cela, considérons v dans V_Σ . Soit un cercle de centre v qui sépare v des autres sommets de G_Σ . Ce cercle intersecte les arêtes $(e_i)_{i \in I}$ de G_Σ incidentes à v . En déformant ce cercle au niveau de ses intersections avec les arêtes de G , nous pouvons obtenir un lacet μ passant par les autres extrémités des arêtes e_i et qui sépare aussi v des autres sommets de G_Σ (figure 8.3).

Ce lacet se décompose en $\nu_1.\nu_2.\dots.\nu_k$ où chaque chemin ν_i ne rencontre G_Σ qu'en ses extrémités, celles-ci étant des sommets de A_Σ .

Chacun de ces chemins est homotope dans Σ_{G_Σ} à un chemin passant par le centre d'une face de G_Σ . Il est donc homotope à la concaténation de deux arêtes de G_Σ^* . Il existe donc un cycle μ^* (pas nécessairement élémentaire) de G_Σ^* homotope à μ . La composante connexe de $\Sigma \setminus \mu^*$ qui contient v ne contient aucun autre sommet de V_Σ . Il existe donc une face de G_Σ^* qui ne contient que v .

Montrons maintenant que chaque face de G_Σ^* contient au moins un sommet de V_Σ . Pour cela, supposons qu'il existe une face f de G_Σ^* qui ne contienne aucun sommet de V_Σ . Comme toutes les arêtes de G_Σ ont une extrémité dans V_Σ et que le bord de la face a une intersection vide avec V_Σ , il n'existe aucune arête de G_Σ incluse dans cette face.

La frontière de l'adhérence de f est un cycle de G_Σ^* . Celui-ci est homotope à zéro dans Σ_{G_Σ} . Il existe donc au moins deux arêtes de G_Σ^* qui sont homotopes, ce qui contredit la définition de G_Σ^* . ■

Lemme 8.43 Soient $G_\Sigma = (V_\Sigma \cup A_\Sigma, E_\Sigma)$ un hyper-graphe plan et $G_\Sigma^* = (F_\Sigma \cup A_\Sigma, E_\Sigma^*)$ son dual. Les ensembles E_Σ et E_Σ^* ont même cardinal.

□ En utilisant la formule d'Euler sur l'hyper-graphe G_Σ , celui-ci étant connexe, nous trouvons :

$$\begin{aligned} |E_\Sigma| + 2 &= |V_\Sigma \cup A_\Sigma| + |F_\Sigma| \\ &= |V_\Sigma| + |A_\Sigma| + |F_\Sigma|. \end{aligned}$$

En procédant de même pour G_Σ^* , nous trouvons :

$$\begin{aligned} |E_\Sigma^*| + 2 &= |F_\Sigma \cup A_\Sigma| + |F_\Sigma^*| \\ &= |F_\Sigma| + |A_\Sigma| + |V_\Sigma|. \quad (\text{d'après la propriété 8.42}) \end{aligned}$$

Le nombre d'arêtes de G_Σ et de G_Σ^* est donc le même. ■

Nous pouvons maintenant donner la démonstration du théorème 8.41.

□ D'après la propriété 8.42, les faces de G_Σ^* sont en bijection avec les sommets de G_Σ . Nous pouvons donc prendre V_Σ comme ensemble des centres des faces de G_Σ^* . Les arêtes de G_Σ sont des chemins d'un sommet face de G_Σ^* à un sommet arête de G_Σ^* . Les arêtes de G_Σ sont donc des candidats possibles pour des arêtes de $(G_\Sigma^*)^*$.

Montrons tout d'abord que E_Σ est inclus dans $(E_\Sigma^*)^*$.

Si ce n'est pas le cas, soient μ et ν deux arêtes de G homotopes dans $\Sigma_{G_\Sigma^*}$. Le lacet $\mu(\nu^{-1})$ forme une courbe de Jordan et comme il est homotope à zéro dans $\Sigma_{G_\Sigma^*}$, il existe une composante connexe Π de $\Sigma \setminus (\mu \cup \nu)$ qui ne contient aucune arête de G_Σ^* . Or Π contient au moins une face de G_Σ ce qui est absurde car cette face est incidente à au moins un sommet arête et donc il existe une arête de G_Σ^* dans Π .

Le lemme 8.43 implique directement que E_Σ et $(E_\Sigma^*)^*$ ont même cardinal. Nous avons donc $E_\Sigma = (E_\Sigma^*)^*$, ce qui implique bien que

$$(G_\Sigma^*)^* = G_\Sigma. \quad \blacksquare$$

8.4.2 Dualité et graphe intermédiaire

Nous présentons ici une propriété fondamentale du graphe intermédiaire : un hyper-graphe plan et son dual ont le même graphe intermédiaire.

Théorème 8.44

Soient $G_\Sigma = (V_\Sigma \cup A_\Sigma, E_\Sigma)$ un hyper-graphe plan et $G_\Sigma^* = (F_\Sigma \cup A_\Sigma, E_\Sigma^*)$ son dual. Nous avons

$$G_I = G_I^*.$$

La démonstration de ce théorème est similaire à celle du théorème de dualité 8.41. Pour le démontrer, commençons par un lemme intermédiaire.

Lemme 8.45 Soit $G_\Sigma = (V_\Sigma \cup A_\Sigma, E_\Sigma)$ un hyper-graphe plan.

Le nombre d'arêtes de G_I et de G_I^* est le même.

□ En utilisant la formule d'Euler sur le graphe G_I , celui-ci étant connexe, nous trouvons :

$$\begin{aligned} |E_I| + 2 &= |V_I| + |F_I| \\ &= |V_\Sigma \cup F_\Sigma| + |A_\Sigma| && \text{(d'après la propriété 8.15)} \\ &= |V_\Sigma| + |F_\Sigma| + |A_\Sigma|. \end{aligned}$$

En procédant de même pour G_I^* , nous trouvons :

$$\begin{aligned} |E_I^*| + 2 &= |V_I^*| + |F_I^*| \\ &= |F_\Sigma \cup V_\Sigma| + |F_I^*| && \text{(d'après la propriété 8.42)} \\ &= |F_\Sigma \cup V_\Sigma| + |A_\Sigma| && \text{(d'après la propriété 8.15)} \\ &= |F_\Sigma| + |V_\Sigma| + |A_\Sigma|. \end{aligned}$$

Le nombre d'arêtes de G_I et de G_I^* est donc le même. ■

Nous pouvons démontrer le théorème 8.44.

□ Les arêtes de G_I sont des chemins dont les extrémités sont un sommet de V_Σ et un sommet de F_Σ . Les arêtes de G_I^* sont donc des candidats possibles pour des arêtes de G_I^* .

Montrons tout d'abord que $E_I \subseteq E_I^*$.

Si ce n'est pas le cas, soient deux arêtes μ et ν de G_I homotopes dans $\Sigma_{G_\Sigma^*}$. Le lacet $\mu \cdot (\nu^{-1})$ forme une courbe de Jordan et comme il est homotope à zéro dans $\Sigma_{G_\Sigma^*}$, il existe une composante connexe Π de $\Sigma \setminus (\mu \cup \nu)$ qui ne contient aucune arête de G_Σ^* . Or Π contient au moins une face de G_I , ce qui est absurde car cette face est incidente à au moins un sommet de F_Σ et donc il existe une arête de G_I^* dans Π .

Comme d'après le lemme 8.45, les graphes G_I et G_I^* ont le même nombre d'arêtes, nous avons bien :

$$G_I^* = G_I. \quad \blacksquare$$

En utilisant ce théorème, nous pouvons transférer des propriétés d'un hyper-graphe plan à son graphe dual.

8.4.3 Deux théorèmes de dualité

Le graphe intermédiaire nous permet de construire certaines décompositions en branches et de relier un hyper-graphe plan et son dual. Ces résultats nous permettent d'obtenir [BMT03] une nouvelle démonstration d'une conjecture de Robertson et Seymour [RS84] démontrée par Lapoire [Lap96]. Celle-ci énonce que la largeur arborescente d'un graphe planaire et celle de son dual diffèrent d'au plus un. Ce résultat est proche d'un théorème de Seymour et Thomas [ST94] qui énonce que la largeur de branches d'un hyper-graphe plan et celle de son dual sont les mêmes. En utilisant notre formalisme, nous sommes en mesure de démontrer ce résultat de la même manière.

Théorème 8.46 ([RS94])

Pour tout hyper-graphe plan G_Σ 2-arête connexe,

$$\text{bw}(G) = \text{bw}(G^*).$$

□ En utilisant le théorème 8.37, nous pouvons considérer une décomposition en branches T_τ de G de largeur minimale induite par une famille \mathcal{F} de courbes de Jordan de G_I . D'après le théorème 8.44, la famille \mathcal{F} induit une décomposition en branches T_{τ^*} de G^* . Chaque courbe de Jordan μ de \mathcal{F} correspond à la fois à une frontière S de T_τ et à une frontière S^* de T_{τ^*} . Les sommets de S sont les sommets initiaux appartenant à μ et ceux de S^* sont les sommets faces de μ . Comme d'après le lemme 8.14, G_I est biparti, les frontières S et S' sont de même taille et donc

$$\text{bw}(G^*) \leq \text{bw}(T_{\tau^*}) = \text{bw}(G).$$

En utilisant le résultat précédent pour G^* , nous obtenons bien :

$$\text{bw}(G) = \text{bw}(G^*). \quad \blacksquare$$

La démonstration pour la largeur arborescente est un peu plus technique mais l'idée générale reste la même.

Théorème 8.47 ([Lap96] et [BMT03])

Pour tout hyper-graphe plan G_Σ fortement arête-connexe,

$$\max(\gamma-1, \text{tw}(G_\Sigma)) - 1 \leq \max(\gamma-1, \text{tw}(G_\Sigma^*)) \leq \max(\gamma-1, \text{tw}(G_\Sigma)) + 1$$

où γ est égal à $\gamma(G_\Sigma)$ et à $\gamma(G_\Sigma^*)$.

□ Soit T_θ une décomposition arborescente de G de largeur minimale qui induit une triangulation minimale H_{T_θ} de G . D'après le théorème 8.39, il existe une décomposition en branches induite par une famille \mathcal{F} de courbes de Jordan de G_I deux à deux parallèles telle que la triangulation induite $H_{\mathcal{F}}$ soit exactement H . Comme d'après le théorème 8.44, le graphe intermédiaire G_I est égal au graphe intermédiaire G_I^* , le corollaire 8.36 permet d'affirmer que la famille \mathcal{F} induit aussi une décomposition en branches de T_{τ^*} de G_Σ^* . Soit T_{θ^*} la décomposition arborescente associée. Les étiquettes de T_{θ^*} sont de deux types : celles des nœuds internes et celles des feuilles.

- si v est un nœud interne de T , d'après la propriété 8.35, les trois frontières bordant v correspondent aux trois courbes de Jordan d'une Θ -structure $\{\nu_1, \nu_2, \nu_3\}$ de \mathcal{F} .

L'ensemble des sommets de G_Σ par lesquels passe au moins l'un des chemins constitue l'étiquette $\theta(v)$ et l'ensemble des faces de G_Σ par lesquelles ils passent constitue l'étiquette $\theta^*(v)$. Le graphe G_I étant biparti, il y a autant de faces que de sommets sur $\nu_1 \cdot (\nu_2^{-1})$. La différence entre les tailles de $\theta(v)$ et de $\theta^*(v)$ vient donc du chemin ν_3 et encore une fois, comme G_I est biparti, cette différence vaut au plus un ;

- si v est une feuille de T , alors la taille $\theta^*(v)$ vaut au plus $\gamma-1$.

Nous obtenons donc :

$$\text{tw}(G_\Sigma^*) \leq \max(\gamma-1, \text{tw}(G_\Sigma)) + 1.$$

Si nous appliquons cette inégalité à l'hyper-graphe dual G_Σ^* , nous obtenons :

$$\max(\gamma-1, \text{tw}(G_\Sigma)) - 1 \leq \max(\gamma-1, \text{tw}(G_\Sigma^*)) \leq \max(\gamma-1, \text{tw}(G_\Sigma)) + 1. \quad \blacksquare$$

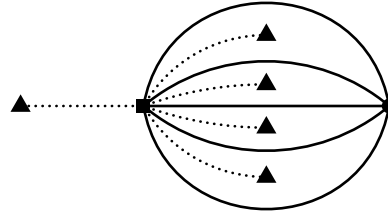
La figure 8.16 illustre l'importance du terme γ .

Théorème 8.48

Pour tout hyper-graphe plan G_Σ ,

$$\max(\gamma-1, \text{tw}(G_\Sigma)) - 1 \leq \max(\gamma-1, \text{tw}(G_\Sigma^*)) \leq \max(\gamma-1, \text{tw}(G_\Sigma)) + 1$$

où γ est égal à $\gamma(G_\Sigma)$ et à $\gamma(G_\Sigma^*)$.



L'hyper-graphe plan G représenté ainsi que son dual G^* n'ont qu'une seule hyper-arête.

La largeur arborescente de G est 0 et celle de G^* est 4.

FIG. 8.16 – Cas où $\text{tw}(G^*) > \text{tw}(G) + 1$

□ D'après le théorème 8.47, nous devons montrer l'inégalité dans le cas où une hyper-arête e de G_Σ en contient une autre e' ou s'il existe une hyper-arête e telle que G_Σ privé des extrémités de e n'est plus connexe. Montrons le théorème par induction sur la taille de G_Σ .

Dans le premier cas, il existe une hyper-arête e en contenant une autre e' . Toute décomposition arborescente de G_Σ est aussi une décomposition arborescente de $G_\Sigma \setminus \{e'\}$. De plus, toute face à laquelle l'hyper-arête e' est incidente est aussi incidente à e . Par conséquent, e'^* est aussi contenue dans e^* et une décomposition arborescente de $(G_\Sigma \setminus \{e'\})^*$ est aussi une décomposition arborescente de G_Σ^* . Comme par induction le théorème est vrai pour $G_\Sigma \setminus \{e'\}$, il l'est aussi pour G_Σ .

Dans le second cas, il existe une hyper-arête e correspondant à un sommet a de G_Σ telle que G privé des extrémités de e ne soit plus connexe. D'après le lemme 8.9, il existe un lacet de G_I qui sépare a du reste des sommets de G_Σ . Considérons un domaine induit par μ_I ne contenant pas a . Son voisinage est une courbe de Jordan μ_I de G_I . D'après le corollaire 8.18, la courbe μ_I induit deux graphes intermédiaires d'hyper-graphes G_1 et G_2 correspondant à des hyper-graphes G_Σ contractés. De même la courbe μ_I induit deux hyper-graphes H_1 et H_2 correspondant à l'hyper-graphe dual G_Σ^* contracté.

Montrons que $\text{tw}(G_\Sigma)$ est égal à $\max(\text{tw}(G_1), \text{tw}(G_2))$. Tout d'abord, si T_θ est une décomposition arborescente de G , en supprimant des étiquettes de T tous les sommets n'apparaissant pas dans G_1 ou dans G_2 , nous obtenons des décompositions arborescentes de G_1 et de G_2 de largeur au plus celle de T_θ . Réciproquement, si $(T_1)_\theta$ et $(T_2)_\theta$ sont des décompositions arborescentes de G_1 et de G_2 , par définition, il existe deux nœuds v_1 et v_2 de T_1 et de T_2 tels que $\theta(v_1)$ et $\theta(v_2)$ contiennent les sommets de $V(\mu_I)$. De plus, les sommets communs à G_1 et G_2 sont des sommets de $V(\mu_I)$; l'arbre obtenu en ajoutant à T_1 et T_2 une arête (v_1, v_2) est donc une décomposition arborescente de T dont la largeur est la plus grande des largeurs de $(T_1)_\theta$ et $(T_2)_\theta$.

Soit e' l'hyper-arête de G_1 et de G_2 correspondant à la partie de G_Σ contractée. Les extrémités de e' sont les sommets de $V(\mu_I)$ et l'hyper-arête duale de e' dans les deux hyper-graphes est l'ensemble des sommets faces par lesquels passe μ_I . Par conséquent, les hyper-graphes G_1^* et G_2^* sont exactement les hyper-graphes H_1 et H_2 . La largeur arborescente $\text{tw}(G_\Sigma^*)$ est donc égale à $\max(\text{tw}(G_1^*), \text{tw}(G_2^*))$ et comme par induction, les largeurs arborescentes $\text{tw}(G_1^*)$ et $\text{tw}(G_2^*)$ sont inférieures à $\max(\gamma-1, \text{tw}(G_1)) + 1$ et $\max(\gamma-1, \text{tw}(G_2)) + 1$, nous avons donc :

$$\max(\gamma-1, \text{tw}(G_\Sigma)) - 1 \leq \max(\gamma-1, \text{tw}(G_\Sigma^*)) \leq \max(\gamma-1, \text{tw}(G_\Sigma)) + 1 \quad \blacksquare$$

Dans le cas des graphes, γ est égal à deux et la formule se simplifie en :

$$\text{tw}(G_\Sigma) - 1 \leq \text{tw}(G_\Sigma^*) \leq \text{tw}(G_\Sigma) + 1.$$

Au cours de cette étude, nous avons adopté une démarche essentiellement combinatoire. L'étude des décompositions en branches nous a permis d'observer qu'une telle décomposition d'un hyper-graphe G est la donnée d'un arbre T et de certains de ses sous-arbres définis à partir de G . La taille d'une arête e de T est alors le nombre de sous-arbres qui contiennent e ; en choisissant T et les sous-arbres associés de sorte que la taille d'une plus grosse arête soit minimale, nous obtenons la largeur de branches de G . Or une telle famille de sous-arbres définit aussi une décomposition arborescente. La taille d'un nœud u de T est le nombre de sous-arbres passant par u moins un et en minimisant la taille d'un nœud, nous obtenons la largeur arborescente de G . Ainsi, certaines décompositions arborescentes et certaines décompositions en branches sont issues d'un même objet arborescent : les matriochkas complètes. La largeur arborescente et la largeur de branches sont obtenues en minimisant respectivement la taille des nœuds et la taille des arêtes de cet arbre.

Cette origine commune suggère que ces deux largeurs possèdent des propriétés similaires et que certaines techniques développées pour l'une peuvent s'adapter à l'autre. De plus, des résultats déjà connus confirment cette idée.

Nous avons donc essayé d'adapter la technique de calcul de la largeur arborescente de Bouchitté et Todinca [BT01b]; celle-ci utilise des triangulations minimales et restreint donc la forme des matriochkas complètes qu'elle considère pour le calcul de la largeur arborescente. Malheureusement, la largeur de branches n'est pas atteinte sur cette sous-famille de matriochkas. Pour remédier à cela, nous avons introduit la notion de profil d'une décomposition en branches grâce auquel nous avons obtenu le théorème de décomposition parallèle. Celui-ci permet de restreindre la forme des matriochkas complètes utilisées pour calculer la largeur de branches. Ces dernières correspondent à des triangulations serrées qui possèdent des propriétés proches de celles des triangulations minimales. À l'aide de ces triangulations particulières, nous avons finalement pu généraliser le résultat de Bouchitté et Todinca et obtenir un algorithme général de calcul de la largeur de branches.

Nous avons appliqué ce nouvel algorithme à certaines classes de graphes, en particulier aux graphes de nombre astéroïde borné ayant un nombre polynomial de séparateurs minimaux et aux graphes d'intervalles circulaires. De cette façon, nous avons pu retrouver et étendre les résultats de Kloks et col [KKM99] sur le calcul de la largeur de branches des graphes d'intervalles.

Les techniques de calcul de la largeur arborescente étaient plus développées que celles de calcul de la largeur de branches; en utilisant notre analogie, nous avons pu obtenir de nouveaux résultats concernant le calcul

de la largeur de branches. Il existe néanmoins une classe d'hyper-graphes pour laquelle la situation est inversée et seule la largeur de branches est connue : les hyper-graphes planaires. Ceci nous a amené à adopter une démarche inverse pour tenter d'adapter des résultats de largeur de branches à la largeur arborescente.

Le calcul de la largeur arborescente des hyper-graphes planaires est hors de portée des techniques combinatoires classiques car ces hyper-graphes possèdent trop de séparateurs minimaux. Pour aborder le calcul de leur largeur de branches, Seymour et Thomas [ST94] ont utilisé un outil combinatoire [RS91] mais surtout des outils topologiques puissants développés par Robertson et Seymour [RS94]. Nous avons réussi à adapter le résultat combinatoire qui fournit une obstruction exacte à la largeur arborescente mais pas les résultats topologiques.

L'étude topologique que nous avons menée nous a néanmoins permis d'obtenir des résultats intéressants. En réintroduisant le graphe intermédiaire, nous avons pu accélérer l'énumération des séparateurs minimaux des hyper-graphes planaires. Ce graphe nous a aussi permis de représenter certaines matriochkas complètes de largeur arborescente et de largeur de branches minimales. En remarquant qu'un graphe planaire et son dual topologique ont le même graphe intermédiaire, nous avons pu démontrer de la même manière qu'ils ont la même largeur de branches et que leurs largeurs arborescentes diffèrent d'au plus un. La démonstration de ce second résultat [BMT03] est plus élémentaire que la démonstration originale de Lapoire [Lap96].

Perspectives

La plupart des techniques de calcul de largeurs de décompositions sont combinatoires ; à partir d'un ensemble de séparateurs minimaux, de cliques maximales potentielles ou de blocs de largeurs de branches connues, elles construisent une décomposition de type donné. De ce point de vue, le calcul de la largeur de branches est plus difficile que celui de la largeur arborescente. En effet, un hyper-graphe peut admettre un nombre de séparations exponentiellement plus grand que son nombre de séparateurs minimaux. C'est par exemple le cas des split-graphs et c'est la raison pour laquelle le problème de la largeur de branches de ces graphes est NP-complet alors que celui du calcul de leur largeur arborescente est linéaire.

Le calcul exact de la largeur arborescente ou de la largeur de branches est un problème NP-complet dans le cas général. Néanmoins, son étude sur des sous-classes a suscité beaucoup de travaux ; une part importante de notre étude est dédiée à ces questions.

À l'heure actuelle, la démarche combinatoire n'utilisant que les séparateurs minimaux ou les cliques maximales potentielles nous semble avoir atteint ses limites pour le calcul de la largeur arborescente. En effet, il est impossible de savoir si un séparateur minimal ou une clique maximale poten-

tielle considéré séparément appartient à une « bonne » décomposition ; ces objets combinatoires ne donnent qu'une information locale sur une décomposition éventuelle d'un graphe. Par conséquent, nous ne pouvons pas restreindre *a priori* l'ensemble des séparateurs minimaux ou des cliques maximales potentielles pour calculer la largeur arborescente. La démarche combinatoire s'applique uniquement à des classes de graphes ayant un nombre polynomial de tels objets, ce qui constitue précisément les résultats de Bouchitté et Todinca [BT01a, BT01b]. Bien que la largeur de branches connaisse les mêmes limites, nous pensons que l'algorithme que nous avons proposé n'est pas encore complètement exploité et qu'il pourrait être appliqué avec succès à d'autres classes de graphes comme celle des graphes de cordes.

Si nous voulons progresser dans le calcul de la largeur arborescente ou, à terme, de la largeur de branches, nous devons trouver des outils qui nous apportent une compréhension globale des décompositions. En utilisant des outils topologiques pour le calcul de la largeur de branches des hyper-graphes planaires, Seymour et Thomas [ST94] ont mis en évidence l'apport de ce type d'outils. Plutôt que de plonger des graphes dans d'autres graphes comme le font les décompositions arborescentes, peut-être devrions-nous profiter du fait que certaines décompositions en branches des hyper-graphes planaires admettent de bonnes représentations comme des familles de cycles pour étudier plus avant le plongement des graphes dans des surfaces de genre quelconque. Nous pouvons nous demander si ces techniques peuvent permettre de calculer la largeur arborescente des graphes planaires.

Comme le calcul exact de la largeur arborescente et de la largeur de branches des hyper-graphes est un problème NP-complet, certains chercheurs se sont tournés vers le calcul approché et le développement d'heuristiques garanties pour le calcul de ces paramètres. Dans ce domaine aussi, l'analogie que nous avons présentée entre décomposition arborescente et décomposition en branches nous paraît prometteuse. Le nombre de frontières est certes plus important que celui de séparateurs minimaux mais une clique maximale potentielle est représentée par exactement trois frontières alors qu'elle est bordée par un nombre *a priori* non borné de séparateurs minimaux. Cette propriété nous paraît d'autant plus intéressante que, dans le cadre d'une heuristique, nous pouvons plus facilement restreindre l'ensemble des frontières utilisées. C'est d'ailleurs exactement ce que font Bouchitté et Todinca [BT03] quand ils construisent une 2-approximation de la largeur arborescente des graphes sans triplets astéroïdes.

Pour finir, la largeur arborescente et la largeur de branches ne constituent que deux paramètres associés aux matriochkas complètes et bien que ces deux paramètres soient très proches, certains problèmes s'expriment mieux en considérant des décompositions arborescentes ou des décompositions en branches. D'autres problèmes peuvent nous conduire à considérer d'autres paramètres comme le diamètre de l'arbre de décomposition et améliorer encore la connaissance des décompositions.

Bibliographie

- [ACP87] Arnborg (S.), Corneil (D.) et Proskurowski (A.), « Complexity of finding embeddings in a k -tree », *SIAM Journal on Algebraic and Discrete Methods*, vol. 8, 1987, p. 277–284.
- [ALS91] Arnborg (S.), Lagergren (J.) et Seese (D.), « Easy problems for tree-decomposable graphs », *Journal of Algorithms*, vol. 12, 1991, p. 308–340.
- [BBC92] Beauquier (D.), Bersetl (J.) et Chrétienne (P.), *Éléments d'algorithmique*. Masson, 1992.
- [BBC00] Berry (A.), Bordat (J.) et Cogis (O.), « Generating all the minimal separators of a graph », *International Journal of Foundations of Computer Science*, vol. 11, n° 3, 2000, p. 397–403.
- [BKK95] Bodlaender (H.), Kloks (T.) et Kratsch (D.), « Treewidth and pathwidth of permutation graphs », *SIAM Journal on Discrete Mathematics*, vol. 8, 1995, p. 606–616.
- [BKKM02] Broersma (H.), Kloks (T.), Kratsch (D.) et Müller (H.), « A generalization of AT-free graphs and a generic algorithm for solving triangulation problems », *Algorithmica*, vol. 33, n° 4, 2002, p. 461–493.
- [BMT03] Bouchitté (V.), Mazoit (F.) et Todinca (I.), « Chordal embeddings of planar graphs », *Discrete Mathematics*, vol. 273, 2003, p. 85–102.
- [Bod96] Bodlaender (H.), « A linear-time algorithm for finding tree-decompositions of small treewidth », *Siam Journal on Computing*, vol. 25, 1996, p. 1305–1317.
- [BPT91] Borie (R.), Parker (R.) et Tovey (C.), « Deterministic decomposition of recursive graph classes », *SIAM Journal on Discrete Mathematics*, vol. 4, 1991, p. 481–501.
- [BT97] Bodlaender (H.) et Thilikos (D.), « Constructive linear time algorithms for branchwidth », dans *Proceedings of the 24th International Colloquium on Automata, Languages and Programming (ICALP'97)*, vol. 1256, p. 627–637, 1997.
- [BT01a] Bouchitté (V.) et Todinca (I.), « Listing all potential maximal cliques of a graph », *Theoretical Computer Science*, vol. 276, n° 1-2, 2001, p. 212–323.
- [BT01b] Bouchitté (V.) et Todinca (I.), « Treewidth and minimum fill-in : grouping the minimal separators », *SIAM Journal on Computing*, vol. 31, 2001, p. 212–232.

- [BT03] Bouchitté (V.) et Todinca (I.), « Approximating the treewidth of at-free graphs », *Discrete Applied Mathematics*, vol. 131, n° 1, 2003, p. 11–37.
- [CLR01] Cormen (T.), Leiserson (C.) et Rivest (R.), *Introduction to algorithms*. MIT Press, 2001.
- [Cou89] Courcelle (B.), « The monadic second-order logic of graphs II : Infinite graphs of bounded width », *Mathematical Systems Theory*, vol. 21, 1989, p. 187–221.
- [CS02] Cook (W.) et Seymour (P.), « Tour merging via branch-decomposition ». Non publié, 2002.
- [Die00] Diestel (R.), *Graph theory*, vol. 173. Springer-Verlag, 2000.
- [Dir61] Dirac (G.), « On rigid circuit graphs », *Abhandlungen Mathematischer Seminare der Universität Hamburg*, vol. 21, 1961, p. 71–76.
- [Esc72] Escalante (F.), « Schnittverbände in graphen », dans *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, vol. 38, p. 199–220, 1972.
- [Fea88] Feautrier (P.), « Parametric integer programming », *Operationnelle/Operations Research*, vol. 22, n° 3, 1988, p. 243–268.
- [Gav74] Gavril (F.), « The intersection graphs of a path in a tree are exactly the chordal graphs », *Journal of Combinatorial Theory*, vol. 16, 1974, p. 47–56.
- [Hal76] Halin (R.), « S -functions for graphs », *Journal Geometry*, vol. 8, 1976, p. 171–186.
- [Hic00] Hicks (I.), *Branch decompositions and their applications*. Thèse de doctorat, Rice University, 2000.
- [HL89] Ho (C.) et Lee (R.), « Counting clique trees and computing perfect elimination schemes in parallel », *Information Processing Letters*, vol. 31, 1989, p. 61–68.
- [KK98] Kloks (T.) et Kratsch (D.), « Listing all minimal separators of a graph », *SIAM Journal on Computing*, vol. 27, n° 3, 1998, p. 605–613.
- [KKM99] Kloks (T.), Kratochvíl (J.) et Müller (H.), « New branch-width territories », dans *Proceedings 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS'99)*, vol. 1563 (coll. *Lecture Notes in Computer Science*), p. 173–183. Springer-Verlag, 1999.
- [Klo96] Kloks (T.), « Treewidth of circle graphs », *International Journal of Foundations of Computer Science*, vol. 7, 1996, p. 111–120.

- [Kru60] Kruskal (J.), « Well-quasi ordering, the tree theorem, and Vászonyi's conjecture », *Transaction of the American Mathematical Society*, vol. 95, 1960, p. 210–255.
- [Lap96] Lapoire (D.), *Structuration des graphes planaires*. Thèse de doctorat, Université Bordeaux I, 1996.
- [LB62] Lekkerkerker (C.) et Boland (J.), « Representation of a finite graph by a set of intervals on the real line », *Fundamenta Mathematicae*, vol. 51, 1962, p. 45–64.
- [Lun90] Lundquist (M.), *Zero Patterns, Chordal Graphs and Matrix Completion*. Thèse de doctorat, Clemson University, 1990.
- [Lya04] Lyaudet (L.), « Largeur de branches des graphes de cordes ». Mémoire de DEA, École normale supérieure de Lyon, 2004.
- [McC03] McConnell (R. M.), « Linear-time recognition of circular-arc graphs », *Algorithmica*, vol. 37, 2003, p. 93–147.
- [PS97] Parra (A.) et Scheffler (P.), « Characterizations and algorithmic applications of chordal graph embeddings », *Discrete Applied Mathematics*, vol. 79, n° 1-3, 1997, p. 171–188.
- [Ros74] Rose (D.), « On simple characterization of k -trees », *Discrete Mathematics*, vol. 7, 1974, p. 317–322.
- [RS84] Robertson (N.) et Seymour (P.), « Graphs minors. III. Planar tree-width », *Journal of Combinatorial Theory Series B*, vol. 36, 1984, p. 49–64.
- [RS86] Robertson (N.) et Seymour (P.), « Graphs minors. II. Algorithmic aspects of tree-width », *Journal of Algorithms*, vol. 7, 1986, p. 309–322.
- [RS91] Robertson (N.) et Seymour (P.), « Graphs minors. X. Obstruction to tree-decomposition », *Journal of Combinatorial Theory Series B*, vol. 52, 1991, p. 153–190.
- [RS94] Robertson (N.) et Seymour (P.), « Graphs minors. XI. Circuits on a surface », *Journal of Combinatorial Theory Series B*, vol. 60, 1994, p. 72–106.
- [RS95] Robertson (N.) et Seymour (P.), « Graphs minors. XIII. The disjoint paths problem », *Journal of Combinatorial Theory Series B*, vol. 64, 1995, p. 240–272.
- [SL97] Shen (H.) et Liang (W.), « Efficient enumeration of all minimal separators in a graph », *Theoretical Computer Science*, vol. 180, 1997, p. 169–180.
- [SSP94] Sundaram (R.), Sher Singh (K.) et Pandu Rangan (C.), « Tree-width of circular-arc graphs », *SIAM Journal Discrete Mathematics*, vol. 7, 1994, p. 647–655.

-
- [ST94] Seymour (P.) et Thomas (R.), « Call routing and the ratcatcher », *Combinatorica*, vol. 14, n° 2, 1994, p. 217–241.
- [Tod99] Todinca (I.), *Aspects algorithmiques des triangulations minimales des graphes*. Thèse de doctorat, École Normale Supérieure de Lyon, 1999.
- [Wag37] Wagner (K.), « Über eine Eigenschaft der ebenen Komplexe », *Mathematische Annalen*, vol. 114, 1937, p. 570–590.

Index

- Θ -structure 119
- adjacence 9
- arbre 10
 - enraciné 10
- arbre de matriochka 46
 - dérivé 54
- arbre des cliques 17
- bloc 21
 - famille complète 60
- branche 45
 - support 45
- capture 75
- chaîne 9
- chemin 99
 - compatible 102
- chevauchement 15, 45
- clique 10
 - maximale 10
 - maximale potentielle 58
- composante
 - connexe 9
 - pleine 10
- corde 9
 - proche d'une composante . 88
- courbe de Jordan 102
 - parallèles 102
- court-circuit 113
- croisement
 - de frontière 76
 - de frontières 13
 - de séparateurs minimaux . 12
- cycle 9
 - élémentaire 9
 - séparant 112
- décomposition arborescente ... 35
 - induite par une matriochka 52
- décomposition en branches 40
 - peignée 77
- découpable 47
- degré 9
- ensemble astéroïde 85
- face 99
- famille complète de blocs 60
- fil
 - textbf 10
- finesse 45
- frontière 13, 75–76
 - d'une arête 40
 - essentielle 76, 76
- graphe 8
 - d*-trapézoïdes circulaires... 94
 - d'incidence 8
 - d'intersection 17
 - d'intervalles circulaires 87
 - de cordes 94
 - de permutation circulaire . 93
 - fortement arête-connexe . 119
 - intermédiaire 106
 - plan 99
 - planaire 100
 - split-graph 73
 - trapézoïdes circulaires 93
- hyper-arête 8
- hyper-graphe 8, 54
 - connexe 9
 - contracté 41
 - d'un bloc 67
 - partiel 9, 50, 51
 - plan 99, 100
 - planaire 100
 - dual 123
 - sous-hyper-graphe 9
 - induit 9

- partiel 9
 sur-hyper-graphe 9
 triangulé 15
- incidence 9
 intervalle d'un séparateur ... 116
 bord 116
 intérieur 116
- lacet 99
 largeur arborescente 38
 d'une matriochka 52
 largeur de branches 40, 43
 d'un bloc 67
 d'une clique 67
 longueur 9
- matriochka 45
 arbre de matriochka 46
 complète 45
 parallèle 45
 modèle d'intersection 17
- nœud 10
 nœud découpable 49
 nombre astéroïde 85
- obstruction 44
- père 10
 paquet 75
 morcelé 76
- parallélisme
 de courbes de Jordan ... 119
 de frontière 76
 de frontières 13
 de séparateurs minimaux . 12
- partie entre séparateurs voisins 21
 point de contrôle 88
 pousser 26
 profil 76
- réalisation d'un séparateur ... 111
 proche d'un sommet 111
 racine 10
 représentation
- compatible 90
 d'un bloc 89
 plane 100
- séparateur 10
 $a, *$ -séparateur minimal ... 30
 a, b -séparateur 10
 a, b -séparateur minimal ... 10
 minimal 10
 ordonné 113
 proche d'un séparateur ... 26
 proche d'un sommet 26
 voisins 21
- séparation 13
 sommet 8
 arête 99
 face 106
 initial 99
- treillis 23
 triangulation 15
 induite par une matriochka 52
 minimale 58
 serrée 60, 81
- voisinage 9