

THÈSE

présentée à

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

Par **Youssou DIENG**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : Informatique

Décomposition arborescente des graphes planaires et routage compact

Soutenue le : 23 octobre 2009

Après avis des rapporteurs :

Pierre Fraigniaud .. Directeur de Recherche
Ioan Todinca Professeur

Devant la commission d'examen composée de :

Bruno Courcelle ..	Professeur	Président
Pierre Fraigniaud .	Directeur de Recherche	Rapporteur
Ioan Todinca	Professeur	Rapporteur
André Raspaud ..	Professeur	Examineur
Cyril Gavoille	Professeur	Directeur de thèse
Stéphane Bessy ...	Maître de conférences	Examineur

Remerciements

Directeur de Thèse : Je voudrais tout d'abord exprimer mes profonds remerciements et ma profonde reconnaissance à *Cyril Gavaille*, qui a dirigé cette thèse dans la continuité de mon stage de Master. Tout au long de ces quatre années, il a su orienter mes recherches aux bons moments et il s'est toujours montré très disponible. Il est toujours prêt à se lancer dans n'importe quel de mes sujets de réflexion. Il a toujours été à mes côtés dans les moments les plus difficiles et il n'a ménagé aucun effort pour me soutenir dans toutes mes activités.

Rapporteurs : Je remercie *Pierre Fraigniaud* et *Ioan Todinca* d'avoir accepté d'être mes rapporteurs. Je les remercie aussi pour la rapidité avec laquelle ils ont lu mon manuscrit et l'intérêt qu'ils ont porté à mon travail.

Commission d'examen : Je remercie aussi les membres du jury qui ont accepté de juger ce travail : *Bruno Courcelle*, *André Raspaud* et *Stéphane Bessy*. Particulièrement, je remercie *André Raspaud* de m'avoir fait aimer la recherche à travers un module d'"initiation à la recherche" au cours de mon Master. Je remercie également *Bruno Courcelle* d'avoir suscité ma curiosité envers les décompositions de graphes à travers un module de Master.

Groupes de travail : Je remercie également *Eric Sopéna* pour son grand soutien dans mes démarches de recherche de financement et aussi de m'avoir initié dans la théorie de graphes.

Je remercie tous les membres des groupes de travail "Graphe et Application" et "Algorithmique Distribuée".

Financement : Je remercie la *fondation d'entreprise banque populaire* d'avoir accepté de financer cette thèse.

Collègues : Je remercie les collègues avec qui j'ai partagé le même bureau pendant toutes ses années à savoir *Joan Mouba* et *Mamadou Moustapha Kanté*.

Je remercie également *Michael Rao* pour son aide dans la relecture et la correction orthographique. Je remercie de même *David Renault* et *Frederic Mazoit* pour leurs conseils, remarques et encouragements.

Je remercie aussi *Maïssa*, *Omer*, *Hedy*, *Hervé*, *Moustapha Diouf*. Je remercie tous les hommes et femmes que j'ai eu à côtoyer de près et de loin au LaBRI parmi lesquels, je peux citer madame *Fabienne Clairand*, *Nicole Lun*, *philippe Biais*, *Katel Guerin*.

Je remercie l'équipe pédagogique du département informatique de l'ENSEIRB et plus particulièrement Denis Lapoire et Robert Cori qui m'ont beaucoup aidé pour ma première année d'enseignement.

Famille : J'adresse un remerciement tout particulier et une dédicace à ma *mère*, mon *père* et à *Imame Mouhamadou Sakhir Gaye* (que la terre leur soit légère). Ma mère m'a toujours donné le surnom du *Docteur de Yeumbeul*.

Je remercie ma famille en commençant par mon "père adoptif" *Papa Youssou Diop*, ma femme *Nafanta* qui a su trouver les mots qu'il faut pour me réconforter et me donner encore plus de motivations dans les moments où le moral est au plus bas. Je dédie cette thèse à mon fils *Mouhamadou Sakhir* né le 27 décembre 2008. Je remercie mes frères et sœurs *Mbaye, Isma, Mody, Téning, Thiabou*, et toute la famille *Ndiobène*.

Je remercie également *Imame Soulaymane Diop* et *Imame Seydina Gaye* de leur soutien sans faille et leurs encouragements sans limite. Je remercie le "dahira" *Ridial, Grand Serigne* et *Ouseynou Laye Kébé*.

Je remercie aussi *Youssou Yade, Lamine Diop, Fatou Ndiaye*, ils ont largement participé à la réalisation de ce projet.

Je dis merci à ma belle famille et particulièrement *Pauline* et sa famille, de même que la famille Ndiaye de Saint-Louis.

Amis : Je remercie aussi *Cheikh Diouf* et sa femme *Khady*, *Amar Sylla* et sa femme *Imane*, *Saly Barry*, *Badou Samb*, *Mansour Diop* et sa femme, *Seydou* et sa femme *Aja*, *Awa Diouf* et son mari, *Issa*, *Babacar*, *Mbaye Ndiaye*, *Paa Abdou Samb*, *Paa Cissé*, *Paa Aly Mboup* et tous les membres du "Khadara".

Décomposition arborescente des graphes planaires et routage compact

Résumé : Savoir comment transmettre une information est fondamental dans un réseau. Il est essentiel que chaque entité du réseau soit capable de décider localement, avec sa vue du réseau, du chemin par lequel l'information doit passer. Ainsi, il est souvent utile d'étudier la topologie du réseau, modélisée par un graphe, pour répondre à ces exigences.

Nous nous intéressons dans un premier temps, à la décomposition arborescente des graphes planaires. En effet, comme dans beaucoup de problèmes de graphes, l'étude de la topologie des graphes nous conduit à procéder à une décomposition du graphe afin d'exploiter les propriétés structurelles qui en découlent.

En suite, nous nous sommes aussi intéressés à la structure des graphes qui excluent un mineur H , en particulier le graphe $K_{2,r}$. Ces travaux nous ont permis d'améliorer les bornes actuelles connues sur la largeur arborescente de ces graphes.

Dans la dernière partie, nous abordons le problème du routage compact. Nous nous sommes intéressés aux schémas de routage de plus courts chemins utilisant des adresses, des tables de routage de tailles optimales de $O(\log n)$ bits, où n est le nombre de sommets du graphe. Nous proposons un tel schéma de routage pour une famille de graphes valués contenant les arbres et les graphes planaire-extérieurs.

Dicipline : Informatique

Mots clefs : Routage compact,
graphe planaire,
graphe planaire-extérieur,
décomposition arborescente,
largeur arborescente,
longueur arborescente,
graphe sans mineur.

LaBRI
Université Bordeaux 1
351 cours de la Libération,
33405 Talence Cedex (FRANCE)

Tree-decomposition of planar graphs and compact routing

Abstract :

In a network, it is crucial to know how to construct an efficient routing scheme. It is fundamental for each entity with its local knowledge of the network, to be able to decide on which link to forward messages. Thus, it is important to study the underlying network topology in order to design routing schemes.

In the first part of this thesis, we construct a new tree-decomposition for planar graphs. In fact, as in many graph problems, the study of the graph structure leads to do a tree-decomposition for exploiting structural properties of the graphs.

In the second part, we have studied the structure of H -minor free graphs, in particular when $H = K_{2,r}$. Our results improve upon previous known bounds about the tree-width of $K_{2,r}$ -minor free graphs.

At last, we treat the problem of compact routing scheme. More precisely, we are interested in shortest-path routing schemes that use $O(\log n)$ bits for addresses, headers and routing tables, where n is the number of vertices in the graph. We propose such a routing scheme for a large family of weighted graphs including outerplanar graphs.

Dicipline : Computer-Science

Keywords : Compact routing,
planar graph,
outerplanar-graph,
tree-decomposition,
tree-width,
tree-length,
minor free graph.

LaBRI
Université Bordeaux 1
351 cours de la Libération,
33405 Talence Cedex (FRANCE)

Table des matières

Table des matières	1
Table des figures	3
Introduction	5
1 Généralités sur les graphes	9
1.1 Quelques notions sur les graphes	9
1.1.1 Les graphes	9
1.1.2 Sous-graphes	13
1.1.3 Les arbres	13
1.2 Mineur de graphe	14
1.3 Décomposition arborescente	16
1.4 Plongement de graphe sur une surface	17
1.4.1 Espace topologique	17
1.4.2 Plongement de graphe sur une surface	18
2 Décomposition arborescente des graphes planaires	19
2.1 Introduction	19
2.2 Décomposition arborescente	21
2.3 Décompositions arborescentes soumises à des conditions particulières	22
2.3.1 Décomposition arborescente où chaque sac possède une partie privée	23
2.3.2 Décomposition arborée	26
2.4 Décompositions arborescentes des graphes planaires	32
2.4.1 Définitions	32
2.4.2 Décomposition arborescente de graphes planaires	34
2.5 Conclusion	47
3 La structure des graphes sans mineur H	49
3.1 Introduction	49
3.2 Présentation des résultats	51
3.3 La structure des graphes sans mineur $K_{2,4}$	52
3.3.1 Préliminaires	52
3.3.2 Démonstration du lemme 3.4 : le cas planaire	53
3.3.3 Démonstration du lemme 3.5 : le cas non-planaire	65

3.3.4	Généralisation du théorème 3.2	68
3.4	La borne supérieure de la largeur arborescente des graphes planaires .	71
3.4.1	Démonstration du théorème 3.3	71
3.4.2	Conclusion	80
4	Schémas de routage compacts de plus courts chemins	83
4.1	Introduction	83
4.1.1	Caractéristiques d'un schéma de routage	83
4.1.2	Présentation des résultats	84
4.2	Routage compact de plus courts chemins dans les graphes planaire- extérieurs non valués	85
4.2.1	Construction du schéma de routage pour les graphes planaire- extérieurs non valués	85
4.2.2	Implémentation	86
4.3	Généralisations	90
4.3.1	Routage compact de plus courts chemins dans les graphes (k, r)-cellulaires	90
4.3.2	Routage compact de plus courts chemins dans les graphes t - feuilletés	107
4.4	Conclusion	113
	Conclusion et perspectives	115
	Index	119
	Bibliographie	123

Table des figures

1.1	Un graphe simple avec 13 sommets et 22 arêtes.	10
1.2	Une grille 9×17	11
1.3	Un graphe G et un séparateur $S = \{3, 10, 9\}$ de G	11
1.4	(a) Un graphe complet à 8 sommets. (b) Un graphe biparti complet $K_{3,3}$ à 6 sommets.	12
1.5	Un sous-graphe couvrant du graphe de la figure 1.1.	13
1.6	Un arbre enraciné sur le sommet étiqueté 1.	14
1.7	Un arbre couvrant de plus courts chemins enraciné sur le nœud 7.	15
1.8	Construction d'un mineur $K_{2,4}$ à partir du graphe de Petersen P_{10}	15
2.1	(a) Un graphe G . (c) Une décomposition arborescente $\mathcal{T}$ de G . (b) Une matérialisation des sacs de $\mathcal{T}$ dans le graphe G	21
2.2	(a) Un graphe G . (b) Une décomposition arborescente $\mathcal{T}$ avec parties privées de G	24
2.3	(a) Un graphe G . (b) Une décomposition arborée $(\mathcal{T}, \mathcal{S})$ de G 1-monotone de profondeur deux.	26
2.4	(a) Une décomposition arborescente enracinée $\mathcal{T}$. (b) Un sac X et 4 sacs Y, Z, W, R adjacents à X dans $\mathcal{T}$ et une représentation des diamants impliquant X	33
2.5	(a) Un arbre $\mathcal{T}$ d'une décomposition arborée $(\mathcal{T}, \mathcal{S})$, enraciné en un sac X adjacent à 4 sacs Y, Z, W, R . (b) Une décomposition arborescente $\mathcal{T}_1$ obtenue après l'application de l'étape 1 de la procédure MAIN1 ($\mathcal{T}_X, X, \mathcal{T}_1 = \emptyset$) à chaque sac X de $\mathcal{T}$	39
2.6	(a) Un arbre $\mathcal{T}$ d'une décomposition arborée $(\mathcal{T}, \mathcal{S})$, enraciné en un sac X adjacent à 4 sacs Y, Z, W, R . (b) Une décomposition arborescente $\mathcal{T}_1$ obtenue après l'application des étape 1 et 2 de la procédure MAIN1 ($\mathcal{T}_X, X, \mathcal{T}_1 = \emptyset$) à chaque sac X de $\mathcal{T}$	40
2.7	(a) Un arbre $\mathcal{T}$ d'une décomposition arborée $(\mathcal{T}, \mathcal{S})$, enraciné en un sac X adjacent à 4 sacs Y, Z, W, R . (b) Une décomposition arborescente $\mathcal{T}_1$ obtenue après l'application de la procédure MAIN1 ($\mathcal{T}_X, X, \mathcal{T}_1 = \emptyset$).	41
3.1	V_8	50
3.2	Un graphe G sans mineur $K_{2,5}$	51
3.3	Un plongement-LMR.	54
3.4	Un mineur $K_{2,4}$ pour la preuve du lemme 3.7.	55
3.5	Les attachements de X à C	56

3.6	Les configurations $A = \{(a), (b)\}$, $B = \{(c), (d)\}$ et $D = \{(e), (f)\}$	58
3.7	Les configurations du cas A pour la règle 3.	59
3.8	Les configurations du cas B pour les règles 4 et 5.	60
3.9	Configurations interdites dans G	62
3.10	Une configuration avec toutes les contraintes si aucune des règles 1 – 6 ne peut s'appliquer.	64
3.11	Dessin planaire de $G_1 = H_0$, G_2 , $G_3 = H_1$, G_4 , et $G_5 = H_2$. Les sommets de degré deux sont colorés en rouge.	69
3.12	Dessin d'un $K_{2,r}$ sur une grille $3 \times r$	72
3.13	(a) Un graphe H dessiné sur une grille $K_{4 \times 4}$. (b) Le dessin du graphe H' construit à partir de H par le lemme 3.15.	74
3.14	(a) Le dessin du graphe H' de la figure 3.14 (a) sur lequel les sommets de niveau i dans H qui possèdent plus de deux voisins dans les niveaux $i - 1$ et $i + 1$ sont représentés chacun par une couleur. (b) Un dessin du graphe H'' construit à partir de H' par le lemme 3.16. Les sommets appartenant au chemin $p_{x_{i,j}}$ induit par le sommet $x_{i,j}$ sont colorés avec la même couleur que $x_{i,j}$. Par exemple le sommet coloré en vert sur la figure (a) induit un chemin de longueur 4 dont tous les sommets de ce chemin sont colorés en vert sur la figure (b)	75
3.15	Un dessin du graphe H''' construit à partir de H''	76
3.16	(a) Une exemple de grille $K_{p \times q}$ avec $p = 3$ et $q = 85$ colonnes et une subdivision de la grille en sous-grilles $A_0, A_1, \dots, A_{t=9}, A_{t+1=10}$ telles que A_0 et A_{10} sont de taille 3×2 et les autres sous-grilles sont de taille 3×9	78
3.17	En (a) et (b) on représente les grilles obtenues après l'étape 4. Les deux grilles sont identiques quelque soit le nombre de colonnes de la sous-grille A_t . On distingue grâce aux couleurs les lignes de la grille $K_{p \times q}$	79
3.18	Reconstruction du mineur $K_{p \times q}$ à partir des grilles (a) et (b) de la figure 3.17. Les zones colorées en noir contiennent les arêtes n'appartenant pas à $K_{p \times q}$. On remarque que pour toute sous-grille A_i et tout couple de lignes L_j, L_k de A_i , il existe une zone noir sur la grille $K_{p \times q}$ entre A_i et A_{i+1} telle que cette zone possède deux chemins p_j, p_k disjoints reliant respectivement les lignes L_j, L_k aux lignes L'_j, L'_k de A_{i+1} . Chaque ligne étant matérialisée par une couleur, il est facile de faire la vérification.	80
4.1	Construction d'un mineur $K_{2,r}$ de G	98
4.2	Construction des tables de routage et adresse.	108
4.3	(a) Une cellule où les sommets colorés forment un segment. (b) Trois cellules formant un graphe 9-feuillet.	109
4.4	(a) Une cellule C et un arbre $\mathcal{T}$ couvrant C enraciné en x où chaque couleur représente un sous arbre de $\mathcal{T}$ et tout ensemble de sommets consécutifs sur la face extérieure de C est représenté par un segment de même couleur que le sous arbre qui le contient. (b) La représentation des segments de C par des sommets ordonnés sur un cercle et le région engendrées par les cordes de $C \cap \mathcal{T}$	112

Introduction

Décomposition arborescente

Les notions de *décomposition arborescente* et de *largeur arborescente* ont été introduites par Robertson et Seymour [RS86] dans leurs travaux sur les *mineurs* de graphe. L'ensemble de ces notions sera proprement défini dans le chapitre 2. Informellement, une décomposition arborescente d'un graphe G est un arbre $\mathcal{T}$ avec des étiquettes sur les nœuds et chaque étiquette est un ensemble de sommets de G . En d'autres termes, la notion de décomposition arborescente permet de représenter un graphe quelconque comme une sorte d'arbre généralisé. La largeur d'une décomposition est le nombre maximum de sommets contenus dans une étiquette diminué de un. La largeur arborescente d'un graphe est le minimum sur toutes les largeurs de toutes les décompositions du graphe. On peut considérer que la largeur arborescente d'un graphe mesure sa similarité par rapport à un arbre ; plus elle est petite, plus ce graphe est "proche" d'un arbre.

La *décomposition arborescente* est une technique couramment utilisée dans la résolution de problèmes combinatoires complexes. En effet, comme nous allons le voir dans le chapitre 2 beaucoup de problèmes NP -difficiles admettent des algorithmes polynomiaux, voir même linéaires pour les classes de graphes de largeur arborescente bornée. Calculer la largeur arborescente ou une décomposition arborescente d'un graphe revient à "découper" ce graphe en sous-graphes, en utilisant certains de ses séparateurs minimaux. Le calcul de la largeur arborescente est NP -difficile. Toutefois, pour tout entier k fixé, il existe un algorithme linéaire pour déterminer si la largeur arborescente d'un graphe est k .

Une *décomposition en chemin* ou *décomposition linéaire* (*path decomposition* en anglais) d'un graphe G est une décomposition arborescente $\mathcal{T}$ de G telle que $\mathcal{T}$ est un chemin. La *largeur de chemin* (*pathwidth* en anglais) est le minimum des largeurs parmi toutes les décompositions en chemin de G .

Routage compact

Le routage point à point de messages est une fonctionnalité fondamentale des réseaux. Les composants d'un schéma de routage sont les tables de routage, les adresses, les en-têtes, et un algorithme de routage. Ce dernier prend comme paramètre l'en-tête du message arrivant et qui, en fonction de la table de routage du sommet courant, détermine le numéro de port (de l'arête) sur lequel le message doit

être retransmis. Le premier en-tête du message est généré par l'émetteur à partir de sa table de routage et de l'adresse de la destination. Dans beaucoup de schémas l'en-tête se résume à l'adresse de destination.

Si la longueur des routes induites par le schéma est clairement un critère de qualité, l'espace requis par les tables de routage, la taille des adresses et des en-têtes, ou encore le temps de latence dans chaque routeur, sont autant de paramètres essentiels qui doivent être optimisés.

Beaucoup d'efforts ont été consacrés à la conception de schémas de routage de bonne qualité en terme de longueur des routes, qui équilibrent la charge de chaque routeur en terme de quantité d'information. L'idée que certains nœuds doivent jouer des rôles différents est une véritable faiblesse. D'autant plus que d'un point de vue théorique, il semble toujours possible de répartir l'information de routage équitablement sur tous les nœuds du réseau tout en gardant la même qualité sur la longueur des routes.

La conception de schémas de routage comportant un compromis taille des tables/longueur des routes passe par l'étude structurelle de la topologie sous-jacente. Par exemple, un des buts recherchés pour les réseaux planaires est de router à la « boussole », sans table de routage, uniquement à l'aide des coordonnées des sommets du réseau. Malheureusement, même pour des structures aussi simples que les triangulations à n sommets cet objectif n'est pas atteignable : pour tout routage de plus courts chemins la taille des tables ou des adresses est au moins de $\Omega(n^{1/4})$ bits pour des triangulations de degré borné, et au moins $\Omega(n^{1/2})$ bits pour les réseaux à croissance bornée (et donc de dimension doublante bornée) [AGGM06].

Pour des structures arborescentes, des solutions avec un espace polylogarithmique existent. Pour les arbres, $O(\log n)$ bits suffisent [FG01b, FG01a, TZ01] et cela s'étend aux graphes de largeur arborescente bornée avec $O(\log^2 n)$ [Pel00]. Cependant la borne est connue pour être optimale seulement dans le cas des arbres [FG02]. Ce résultat pour les arbres est particulièrement intéressant en théorie du routage compact. Par exemple, le résultat quasi-optimal de Thorup et Zwick [TZ01] pour les graphes quelconques est très dépendant du routage dans les arbres. Il énonce que tout graphe possède un schéma de routage avec des tables de taille approximativement $n^{1/k}$ par sommet garantissant des routes de longueur au plus $O(k)$ fois la distance, pour tout paramètre entier $k \geq 1$.

Fraigniaud et Gavoille [FG02] ont posé explicitement la question de savoir si le schéma optimal des arbres pouvait être étendu à d'autres graphes, en particulier est-ce que tout graphe planaire-extérieur possède un schéma de routage de plus courts chemins avec des tables et des adresses de $O(\log n)$ bits ? La meilleure borne connue étant de $O(\log^2 n)$ bits [Pel00].

Plan de la thèse

Dans toute la thèse, le paramètre n représentera le nombre de sommets du graphe considéré.

Chapitre 1 : Généralités

Dans ce chapitre, nous introduisons les différentes notions utilisées en théorie des graphes de manière courante et dont nous nous servons dans ce document.

Nous y présentons également les notions liées aux mineurs de graphe et de décomposition arborescente, concepts fondamentaux et au centre de nos travaux.

Chapitre 2 : Décomposition arborescente dans les graphes planaires

Dans ce chapitre nous présentons un algorithme polynomial qui, pour tout graphe planaire de longueur arborescente ℓ , donne une décomposition arborescente $\mathcal{T}$ dans laquelle chaque sac est composé d'au plus 12 plus courts chemins de longueur ℓ . Chaque sac de $\mathcal{T}$ possède au plus 6 chemins de longueur au plus $3\ell/2$, composés chacun d'au plus deux plus courts chemins. Ce résultat met en évidence pour la première fois une relation entre la largeur arborescente et la longueur arborescente des graphes planaires. Une partie des résultats ont donné lieu à une publication [DG09a].

Chapitre 3 : La structure des graphes sans mineur $K_{2,4}$

Dans cette partie, nous nous intéressons à la structure des graphes ne contenant pas comme mineur un graphe fixé H . De nombreuses caractérisations sont connues lorsque H contient au plus 5 sommets, en particulier pour $H = K_5$ [Wag37]. Le problème est beaucoup plus difficile lorsque H contient 6 sommets, voire plus. Par exemple, la caractérisation des graphes sans mineur K_6 est non résolue, et est en relation avec la conjecture d'Hadwiger concernant le nombre chromatique des graphes sans K_r (voir [Kaw07] pour une plus ample discussion sur l'importance des graphes sans mineur K_6). Citons également la conjecture de Jørgensen [Jør94] disant que tout graphe 6-connexe G sans mineur K_6 possède un sommet u tel que $G \setminus \{u\}$ est planaire.

Nous prouvons le théorème 3.2 disant que tout graphe 2-connexe G sans mineur $K_{2,4}$ possède deux sommets u, v tels que $G \setminus \{u, v\}$ est planaire-extérieur.

En particulier, les graphes sans mineur $K_{2,4}$ sont de largeur arborescente au plus 4, améliorant un résultat de Bodlaender *et al.* [BKK95a] qui est 6. Cette borne est optimale à cause de K_5 . En fait, nous construisons un algorithme de complexité linéaire qui, pour tout graphe 2-connexe soit exhibe un mineur $K_{2,4}$ soit détermine deux sommets dont la suppression rend le graphe planaire-extérieur.

Le théorème précédent ne peut pas être étendu trivialement pour les graphes sans mineurs $K_{2,r}$ avec $r > 4$. En effet, nous montrons que pour tout entier k , il existe un graphe 2-connexe sans mineur $K_{2,5}$ pour lequel il est nécessaire d'enlever k sommets pour le rendre planaire-extérieur.

Nous montrons aussi le théorème 3.3 disant que si H est un graphe dessinable sur une grille $p \times q$, alors tout graphe G sans mineur H , est de largeur arborescente $O(p^{3/2}\sqrt{q})$. En conséquence de ce résultat, nous montrons que les graphes sans mineur $K_{2,r}$ sont de largeur arborescente $O(\sqrt{r})$. Ce qui améliore la borne actuelle

due à Thilikos, qui est de $r + 2$. Une version préliminaires de ces résultats a été présentée à un colloque [DG08].

Chapitre 4 : Schémas de routage compact de plus courts chemins

Dans la section 4.2 nous nous intéressons aux routage de plus courts chemins dans les graphes planaire-extérieurs non valués. Un graphe planaire extérieur est un graphe pouvant être plongé sur le plan sans croisement d'arête et de sorte que tous les sommets soient sur le bord de la face extérieure.

Nous avons en fait mis en place un schéma de routage de plus courts chemins pour les graphes planaire-extérieurs, avec des tables de routages, des adresses et des en-têtes de $O(\log n)$ bits. La méthode utilisée pour mettre en place ce schéma sera utilisée pour la construction du schéma de routage développé dans la section 4.3.1.

Ce schéma de routage est une extension du résultat optimal de Fraigniaud et Gavoille [FG01b] et indépendamment Thorup et Zwick [TZ01] dans les arbres. Fraigniaud et Gavoille [FG02] avaient posé la question de savoir si leur résultat pouvait être étendu à d'autres familles de graphes telles que par exemple les graphes planaire-extérieurs.

Dans la section 4.3 nous généralisons le schéma de routage dans les graphes planaire-extérieurs à d'autres familles de graphes plus générales. Plus précisément, dans la section 4.3.1 nous présentons un schéma de routage de plus courts chemins pour les graphes (k, r) -cellulaires, c'est-à-dire les graphes ne contenant pas $K_{2,r}$ comme mineur et dont les composantes bi-connexes peuvent être rendues planaire-extérieurs par la suppression de k sommets.

Nous avons mis en place un schéma de routage de plus courts chemins avec des adresses et des en-têtes de $O(k \log n)$ bits et des tables de routage de $O(kr \log n)$ bits. Le temps de routage est de $O(\log k + \log r)$ par sommet. Ce schéma de routage étend le résultat optimal dans les arbres [FG01b, TZ01] aux graphes (k, r) -cellulaires pour k et r bornés. En particulier pour les graphes sans mineur $K_{2,4}$, nous avons montré dans le chapitre 3 qu'ils étaient $(2,4)$ -cellulaires, et donc ont un schéma de routage de plus courts chemins avec des tables et des adresses de taille $O(\log n)$ bits. La borne connue pour ces graphes était due à Peleg [Pel00], elle est de $O(\log^2 n)$ bits. Une partie de ces résultats a été publiée dans [DG09b].

Dans la section 4.3.2 nous présentons un schéma de routage de plus courts chemins dans les graphes t -feuilletés. Un graphe t -feuilleté G est un graphe pouvant être partitionné en un ensemble de sous-graphes de G tel que chaque arête de G soit contenue dans au moins un sous-graphe et il y a au plus t sommets appartenant à la fois à plusieurs sous-graphes avec leur multiplicité.

Nous avons en fait mis en place un schéma de routage de plus courts chemins avec des tables de $O(t \log n)$ bits par sommets, des adresses et des en-têtes de $O(\log n)$ bits. Le temps de décision est de $O(\log t)$ par sommet. Ce résultat a fait l'objet d'une publication dans [DG07a].

Chapitre 1

Généralités sur les graphes

Les graphes permettent de modéliser toute situation dans laquelle il y a des interactions entre les objets. On peut donner comme exemples les réseaux routiers, les réseaux de chemin de fer, les réseaux sociaux ou encore le réseau internet. Les techniques utilisées en théorie de graphes permettent de répondre à beaucoup de problèmes algorithmiques posés sur ces réseaux. En effet, étudier les propriétés de ces réseaux revient à étudier les propriétés structurelles de leurs topologies représentées par des graphes. Une question typique qu'on pourrait se poser dans un réseau routier est quel est un plus court chemin reliant une ville A à une ville B . Une des techniques couramment utilisées pour résoudre à des problèmes algorithmiques sur les graphes est la décomposition de graphe.

Dans ce chapitre, nous introduisons les différentes notions utilisées en théorie des graphes de manière courante et dont nous nous servons dans ce document.

Nous y présentons également des notions moins classiques telles que les notions liées aux mineurs de graphe, et de décomposition arborescente, concepts fondamentaux et au centre de nos travaux. Pour plus de détails, nous vous conseillons de vous référer aux travaux de Robertson et Seymour [RS86].

Nous présentons aussi dans la dernière partie de ce chapitre les notions de plongement de graphe sur une surface. Ces notions seront utiles pour la compréhension de certaines parties des chapitres 3 et 4.

1.1 Quelques notions sur les graphes

Les notions présentées dans cette partie sont donc probablement déjà connues des lecteurs avertis. Pour plus de détails, nous conseillons au lecteur de se référer au livre de Claude Berge [Ber83].

1.1.1 Les graphes

Un *graphe* G est un couple (V, E) où V est l'ensemble des *sommets* et E l'ensemble de *arêtes*. Toute arête $e \in E$ correspond à une paire $\{u, v\} \in V$ de sommets représentant ses *extrémités*. Une arête (u, v) est une boucle si $u = v$.

Dans toute la suite de cette thèse, nous allons considérer que des graphes non orientés, sans boucle ni arête multiple c'est à dire des graphes *simples*, (voir figure

1.1). On notera par n le nombre de sommets du graphe et m le nombre d'arêtes. Le nombre de sommets n est aussi appelé *l'ordre* du graphe.

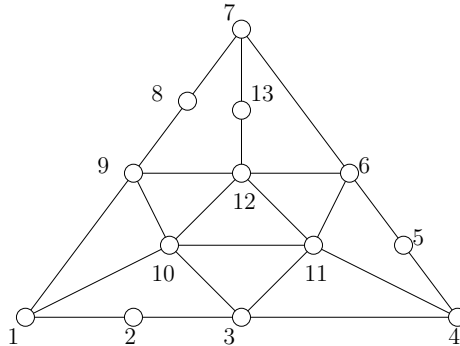


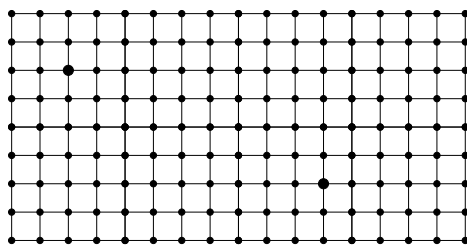
FIG. 1.1: Un graphe simple avec 13 sommets et 22 arêtes.

Deux sommets u et v sont dit *adjacents* s'il existe une arête entre u et v dans G . On dit aussi que u et v sont *voisins*. L'ensemble des sommets adjacents à u est appelé le voisinage de u ; il sera noté par $N(u)$. Le degré de u , noté $\deg(u)$, désigne le nombre de voisins de u .

Un chemin dans un graphe : Un *chemin* dans un graphe $G = (V, E)$ est une suite $(u_0, u_1, \dots, u_k)$ de sommets telle que deux sommets consécutifs v_i et v_{i+1} sont reliés par une arête. La longueur d'un tel chemin est le nombre d'arêtes qui le composent; k dans notre exemple. Le chemin est dit *simple* si tous les sommets sont distincts; on dit aussi que ce chemin est une *chaîne*. Un chemin tel que le sommet de départ v_0 est égal au sommet d'arrivée v_k est un *cycle*. Un *plus court chemin* entre deux sommets u et v est un chemin simple de longueur minimale. On autorise qu'il y ait des poids (valuations) sur les arêtes. Dans ce cas on dit que le graphe est *valué* et tout plus court chemin entre deux sommets u et v est un chemin de poids minimum entre u et v . La *distance* dans un graphe G entre deux sommets u et v notée $\text{dist}_G(u, v)$ est la longueur d'un plus court chemin dans G entre u et v . Le *diamètre* d'un graphe G noté $\text{diam}(G)$ est la plus grande distance entre deux sommets quelconques de G . De manière formelle on a : $\text{diam}(G) = \max_{x, y \in V} \{\text{dist}_G(x, y)\}$

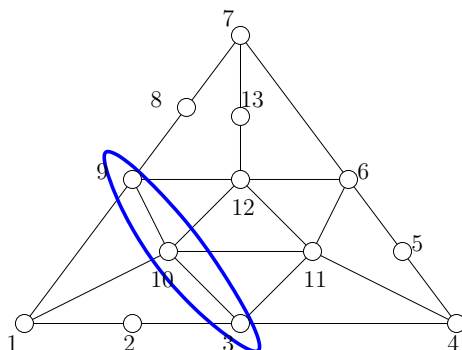
Une grille : Une grille $K_{p \times q}$ est un graphe composé de $p \times q$ sommets $v_{i,j}$ avec $1 \leq i \leq p, 1 \leq j \leq q$, dans lequel deux sommets $v_{i,j}$ et $v_{k,l}$ sont adjacents si $|i - k| + |j - l| = 1$ (cf. figure 1.2). Où p représente le nombre de lignes et q le nombre de colonnes.

Subdivision de graphe *Subdiviser une arête* dans un graphe revient à remplacer cette arête par une chaîne de sommets intermédiaires. Une *subdivision d'un graphe* G est un graphe H obtenu en subdivisant les arêtes de G .

FIG. 1.2: Une grille 9×17 .

Graphe connexe : Un graphe est dit *connexe* si pour toute paire de sommets distincts, il existe un chemin qui les relie. Un *séparateur* dans un graphe connexe est un ensemble de sommets tels que leur suppression engendre un graphe non connexe. Si S est un séparateur de G , alors il existe deux sommets a et b de G qui sont dans deux composantes connexes différentes de $G \setminus S$. On dit aussi que S est un *ab-séparateur*. Un *ab-séparateur* S de G est dit minimal si aucun sous ensemble de S n'est un *ab-séparateur* de G . Un séparateur S de G est dit minimal si tout séparateur S' de G est tel que $|S'| \geq |S|$. Par exemple sur la figure 1.3 le séparateur $S = \{3, 9, 10\}$ déconnecte le graphe en deux composantes connexes dont l'une A contient les sommets $\{1, 2\}$ et l'autre B contient les sommets $\{4, 5, 6, 7, 8, 11, 12, 13\}$. Le séparateur S est aussi un séparateur minimal de G .

Graphe k -connexe : Un graphe G est dit *k -connexe* s'il faut supprimer au moins k sommets pour le rendre non connexe ; en d'autres termes tout séparateur de G est de taille au moins k . La connectivité k_G de G est le nombre minimum de sommets dont l'élimination dis-connecte G . La connectivité de G est donc la taille du plus petit *ab-séparateur* minimal de G . Un *sommet d'articulation* d'un graphe est un sommet séparateur du graphe. Notons aussi que le terme *bi-connexe* est souvent utilisé à la place de 2-connexe.

FIG. 1.3: Un graphe G et un séparateur $S = \{3, 10, 9\}$ de G .

Grphe complet : Un graphe est dit *complet* si pour toute paire $\{u, v\}$ de sommets, il existe une arête dont les extrémités sont u et v . Un graphe complet est donc un graphe dont le diamètre est égal à 1. Un graphe complet à n sommets noté K_n est aussi appelé une clique. Un exemple de graphe complet à 8 sommets est donné sur la figure 1.4 (a).

Si G n'est pas une clique, il existe deux sommets u et v non adjacents, dont $V \setminus \{u, v\}$ est un ensemble dont l'élimination dis-connecte G , et par conséquent :

$$k_G \leq |V \setminus \{u, v\}| = n - 2.$$

Si G est une clique d'ordre n , on a $k_G = n - 1$.

Pour tout graphe G , si G n'est pas une clique et si $\mathbb{S}$ est la famille d'ensembles séparateurs de G , G est k -connexe si et seulement si $k_G = \min_{S \in \mathbb{S}} \{|S|\} \geq k$

Grphe biparti : Un graphe est dit *biparti* s'il existe une partition de son ensemble de sommets en deux sous-ensembles U et V telle que chaque arête ait une extrémité dans U et l'autre dans V . Un graphe est biparti si et seulement si il ne contient donc pas de cycle de longueur impaire. Un graphe biparti est dit *biparti complet* si chaque sommet de U est relié à chaque sommet de V (cf. figure 1.4 (b)). On le note $K_{|U|, |V|}$. Le graphe biparti complet qui est le plus utilisé dans cette thèse est $K_{2,4}$. On appellera pôles les deux sommets de degré quatre.

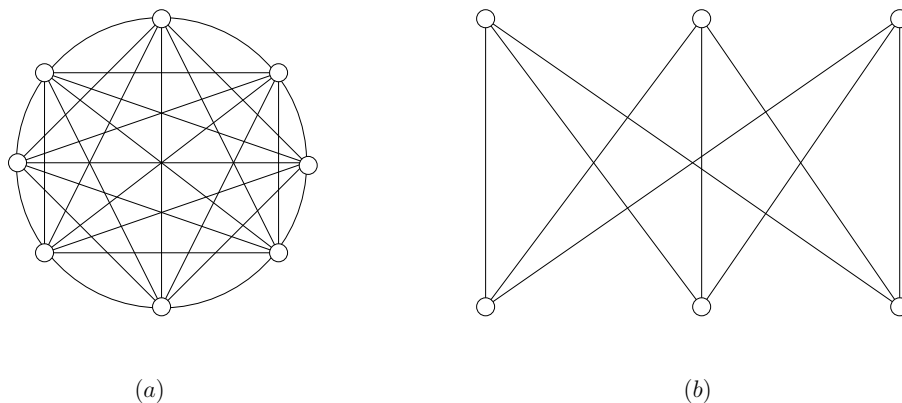


FIG. 1.4: (a) Un graphe complet à 8 sommets. (b) Un graphe biparti complet $K_{3,3}$ à 6 sommets.

Le théorème suivant dû à Kuratowski a permis de donner une caractérisation des graphes planaires.

Théorème 1.1 [Kur30] *Un graphe G est planaire si et seulement si G ne contient pas (comme sous-graphe partiel) une subdivision de K_5 ni une subdivision de $K_{3,3}$.*

Homomorphisme de graphes : Un homomorphisme d'un graphe G vers un graphe H est une application h de $V(G)$ vers $V(H)$ qui préserve les arêtes. En

d'autres termes, pour toute arête (u, v) de G , il existe une arête $(h(u)h(v))$ dans H avec $h(u)$, l'image par h de u et $h(v)$, l'image par h de v .

1.1.2 Sous-graphes

Un *sous-graphe* d'un graphe $G = (V, E)$ est un graphe $G' = (V', E')$ tel que $V' \subseteq V$ et $E' \subseteq E$. Le graphe G est appelé un *super-graphe* de G' .

Sous-graphe induit : Un graphe $G' = (V', E')$ est un *sous-graphe induit* d'un graphe $G = (V, E)$ si pour tout couple de sommets $(x, y) \subseteq V'$ tel que $(x, y) \in E$ alors on a $(x, y) \in E'$. En d'autres termes, si deux sommets de V' sont adjacents dans G , alors ils sont adjacents dans G' .

Sous-graphe couvrant : Un graphe $G' = (V', E)$ est un *sous-graphe couvrant* d'un graphe $G = (V, E)$ si $V' = V$. En d'autres termes un sous-graphe couvrant de G contient tous les sommets de G . Un exemple de sous-graphe couvrant du graphe de la figure 1.1 est donné sur la figure 1.5. Les arêtes du sous graphe sont représentées en rouge. Ce graphe n'est pas connexe car il n'y a pas de chemin entre 7 et 3.

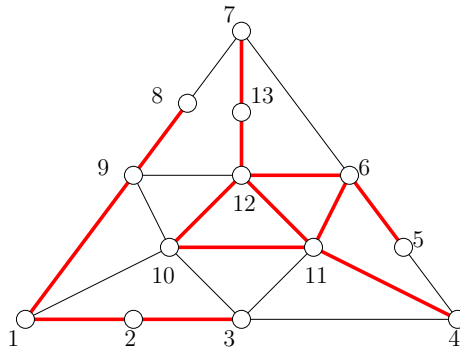


FIG. 1.5: Un sous-graphe couvrant du graphe de la figure 1.1.

Pour tout graphe G et un ensemble X de sommets de G , le sous-graphe de G induit par X sera noté $G[X]$.

1.1.3 Les arbres

Un *arbre* est par définition un graphe connexe sans cycle. Les sommets d'un arbre sont généralement appelés les nœuds de l'arbre. Un arbre peut aussi être défini comme un graphe connexe tel que pour tout couple de sommets (u, v) , il existe une unique chaîne entre u et v .

Arbre enraciné : Un arbre est dit *enraciné* s'il admet un sommet distingué appelé *racine*. Il est aussi appelé une *arborescence*. Les arbres enracinés sont souvent représentés de telle sorte que la racine soit le sommet situé le plus haut (cf.

figure 1.6). Une *forêt* est un graphe dont chaque composante connexe est un arbre.

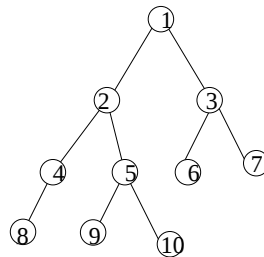


FIG. 1.6: Un arbre enraciné sur le sommet étiqueté 1.

On dit qu'un nœud v est *descendant* d'un nœud u si le nœud u est sur le chemin entre v et la racine de l'arbre. On dit aussi que u est un *ancêtre* de v . Si u et v sont reliés par une arête alors on dit que u est le *père* (ou *parent*) de v et donc v est un enfant de u . Deux nœuds ayant le même père sont dit *frères*. Tout nœud autre que la racine qui possède un ou plusieurs enfants est un nœud *interne*. Inversement, tout nœud qui ne possède pas d'enfant est une *feuille*. La *profondeur* d'un nœud est la distance qui le sépare de la racine. La profondeur de l'arbre est la plus grande profondeur de ses feuilles. On appelle plus petit ancêtre commun entre u et v dans un arbre T , noté $nca_T(u, v)$ le nœud ancêtre de u et v ayant la plus grande profondeur.

Parcours en profondeur : Un *parcours en profondeur* (en anglais Depth First Search DFS) d'un arbre T consiste à traiter la racine r de T , puis parcourir récursivement les sous arbres issus des enfants de r .

Parcours en largeur : Un *parcours en largeur* (en anglais Breadth First Search BFS) d'un arbre T consiste à traiter les nœuds par niveau de profondeur : tous les nœuds de niveau i seront traités avant tout nœud de niveau $i + 1$. Faire un parcours en largeur d'un graphe consiste à choisir un sommet, traiter ce sommet ensuite traiter les sommets à distance 1, ensuite les sommets à distance 2 et ainsi de suite.

Arbre couvrant : Un sous-graphe couvrant G' de G est un *arbre couvrant* G si G' est un arbre. Un arbre G' couvrant G est dit de plus courts chemins s'il existe un sommet r tel que pour tout sommet v , $\text{dist}_G(r, v) = \text{dist}_{G'}(r, v)$. Un tel arbre peut être obtenu à partir de tout graphe G en faisant un BFS sur G . Le graphe de la figure 1.7 est un arbre couvrant du graphe de la figure 1.1.

1.2 Mineur de graphe

Soit (x, y) une arête d'un graphe $G = (V, E)$. On note par $G \setminus (x, y)$ le graphe obtenu à partir de G en contractant l'arête (x, y) et en supprimant toutes les boucles

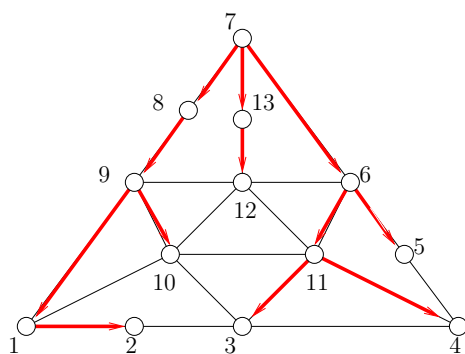
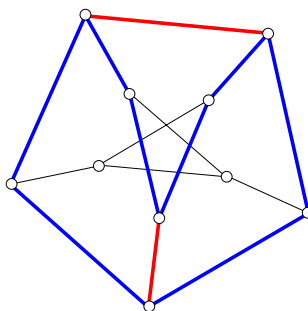


FIG. 1.7: Un arbre couvrant de plus courts chemins enraciné sur le nœud 7.

et les arêtes parallèles résultantes. On note par $G > H$ si un graphe isomorphe à H peut être obtenu à partir d'un sous graphe de G par contraction d'arête. On dit dans ce cas que H est un mineur de G . Donc, si H est un mineur de G , alors G admet un sous graphe G' tel que $V(G')$ peut être partitionné en $|H|$ sous-ensembles disjoints et connexes dans G , indicés par les sommets de H et il y a au moins une arête entre deux sous-ensembles associés à deux sommets adjacents de H . Un graphe G est dit sans mineur H , si le graphe H n'est pas un mineur de G .

Par exemple, la figure 1.8, représente le graphe de Petersen P_{10} . En contractant les deux arêtes rouges et en ne gardant que les arêtes bleues, et les sommets adjacents aux arêtes bleues, on obtient un $K_{2,4}$. Donc le graphe de Petersen admet $K_{2,4}$ comme mineur.

FIG. 1.8: Construction d'un mineur $K_{2,4}$ à partir du graphe de Petersen P_{10} .

Théorème 1.2 [RS95] *Pour tout graphe fixé H et pour tout graphe G d'ordre n , il existe un algorithme qui en $O(n^3)$ décide si un graphe G contient H comme mineur.*

Par ailleurs, nous avons le théorème suivant qui est une reformulation par Wagner du théorème 1.1 :

Théorème 1.3 [Wag37] *Un graphe G est planaire si et seulement si G ne contient ni K_5 ni $K_{3,3}$ comme mineur.*

Un graphe sans mineur $K_{2,3}$ et K_4 est dit *planaire-extérieur*.

Mineur topologique : Un graphe H est un mineur topologique d'un graphe G si G contient un sous-graphe isomorphe à un graphe pouvant être obtenu à partir de H par une subdivision d'arêtes. Dans ce cas, nous pouvons aussi dire que G contient une subdivision de H .

1.3 Décomposition arborescente

Les notions de *décomposition arborescente* et de *largeur arborescente* ont été introduites par Robertson et Seymour [RS86] dans leurs travaux sur les *mineurs* de graphe. L'ensemble de ces notions sera proprement défini dans le chapitre 2. La notion de décomposition arborescente permet de représenter un graphe quelconque comme une sorte d'arbre généralisé. Informellement, une décomposition arborescente d'un graphe G est un arbre $\mathcal{T}$ avec des étiquettes sur les nœuds et chaque étiquette est un ensemble de sommets de G . La largeur d'une décomposition est le nombre maximum de sommets contenus dans une étiquette diminué de un. La largeur arborescente d'un graphe est le minimum sur toutes les largeurs de toutes les décompositions du graphe. On peut considérer que la largeur arborescente d'un graphe mesure sa similarité par rapport à un arbre ; plus elle est petite, plus ce graphe est "proche" d'un arbre.

La *décomposition arborescente* est une technique couramment utilisée dans la résolution de problèmes combinatoires complexes. En effet, comme nous allons le voir dans le chapitre 2 beaucoup de problèmes *NP*-difficiles admettent des algorithmes polynomiaux, voir même linéaires pour les classes de graphes de largeur arborescente bornée. Calculer la largeur arborescente ou une décomposition arborescente d'un graphe revient à "découper" ce graphe en sous-graphes, en utilisant certains de ses séparateurs minimaux. Le calcul de la largeur arborescente est *NP*-difficile. Toutefois, pour tout entier k fixé, il existe un algorithme linéaire pour déterminer si la largeur arborescente d'un graphe est k .

Les graphes sans mineur K_4 sont de largeur arborescente 2. Par ailleurs, toute subdivision de K_4 contient $K_{2,3}$ comme mineur. Les graphes planaire-extérieurs sont donc de largeur arborescente au plus 2.

Si $\mathcal{F}$ est une famille de graphes et $G \in \mathcal{F}$, si tout mineur de G est dans $\mathcal{F}$, alors on dit que $\mathcal{F}$ est une famille *close par mineurs*. Par exemple, la famille des graphes planaires est une famille close par mineurs. On dit qu'un graphe H est une *obstruction* pour une famille de graphes $\mathcal{F}$ si $H \notin \mathcal{F}$ et si tout mineur de H appartient à $\mathcal{F}$. Dans leur travaux sur les mineurs Robertson et Seymour [RS03] montrent que pour toute famille close par mineurs, ses obstructions sont en nombre fini. Ces notions seront d'avantage détaillées dans le chapitre 2.

1.4 Plongement de graphe sur une surface

Cette partie a pour objectif de familiariser le lecteur aux notions de surface et de plongement de graphe dans une surface. Pour plus de détails, nous conseillons aux lecteurs de se référer au livre de Mohar et Thomassen [MT01]. Ces deux notions seront utiles pour la compréhension de certaines parties de cette thèse. Pour comprendre la notion de surface avons besoin de comprendre ce qu'est un espace topologique.

1.4.1 Espace topologique

On appelle *espace topologique* tout couple constitué par un ensemble E et un ensemble O de parties de E appelées *ensembles ouverts* (ou *ouverts* en abrégé) et satisfaisant les trois propositions suivantes :

1. Toute réunion (finie ou non) d'ouverts est un ouvert.
2. Les ensembles E et $\emptyset$ sont des ouverts.
3. Toute intersection finie d'ouverts est un ouvert.

On dit que l'ensemble O de parties de E définit sur E une topologie.

Une surface S est un espace topologique d'Hausdorff connexe, compact dans lequel chaque point possède un voisinage homomorphe au plan euclidien $\mathbb{R}^2$. Si S est un espace topologique, alors tout arc, tout cercle et tout disque de S est homomorphe dans S , respectivement à l'intervalle réel $[0, 1]$, au cycle unitaire $S^1 = \{x \in \mathbb{R}^2 : \|x\| = 1\}$, et au disque unitaire $\{x \in \mathbb{R}^2 : \|x\| \leq 1\}$ ou $\{x \in \mathbb{R}^2 : \|x\| < 1\}$. Un espace topologique S est dit connexe par arc si toute paire de points de S peut être relié par un arc dans S . Deux points de S sont équivalents s'ils peuvent être reliés par un arc dans S . Ainsi, chaque ensemble connexe de points de S induit une classe d'équivalence. Les classes d'équivalences sont aussi appelées des régions de S . Si X est un sous-ensemble de points de S , alors la frontière de X est l'ensemble de tous les points y de S tels que tout voisinage de y intersecte à la fois X et $S \setminus X$. La frontière F de X sépare X et $S \setminus X$. Si une région C de la frontière de X est un cycle, alors on dit que C est un cycle englobant X . Il existe un théorème fondamental qui dit que toute surface peut être obtenue à partir de la sphère $S^2 = \{x \in \mathbb{R}^3 : \|x\| = 1\}$ en ajoutant un nombre fini de d'anses ou un nombre fini d'anses tordues. De plus, ce théorème montre que si deux surfaces S_0 et S_1 n'ont pas le même nombre d'anses ou d'anses tordues, alors S_0 et S_1 sont deux surfaces différentes.

Pour ajouter une anse à une surface S , on considère une région R localement homomorphe à un plan. Ensuite, on retire de R deux disques ouverts disjoints de même diamètre et on oriente la frontière de chaque disque de telle sorte que les deux orientations soient différentes dans le plan défini par P et on ajoute l'anse.

Pour ajouter une anse tordue à une surface S , on considère une région R localement homomorphe à un plan. Ensuite, on retire de R deux disques ouverts disjoints de même diamètre et on oriente la frontière de chaque disque de dans le même sens dans le plan R et on ajoute l'anse.

Toute surface $\mathbb{S}_g$ obtenue en ajoutant g anses à une sphère est appelé surface *orientable de genre g* et toute surface $\mathbb{N}_{\tilde{g}}$ obtenue en ajoutant $\tilde{g}$ anses tordues à une sphère est appelé surface *non-orientable de genre $\tilde{g}$* .

1.4.2 Plongement de graphe sur une surface

Plonger un graphe G sur une surface S est une fonction $\sigma : G \mapsto S$ telle que les sommets de G ont des images disjointes de S et que chaque arête xy de G a comme image une courbe simple (ou arc) reliant les points $\sigma(x), \sigma(y)$ (les images respectives de x et y dans S) dans S . De plus, deux arêtes disjointes de G ont des images disjointes dans S et deux arêtes s'intersectant dans G en un sommet u s'intersectent dans S en $\sigma(u)$. Le plongement de G noté $\sigma(G)$ est l'ensemble des images de tous les sommets et arêtes de G dans S . Une face F de G dans S est une région R de $S \setminus \sigma(G)$ et le sous graphe de G ayant pour image dans S la frontière de R est le contour de F .

Le plan Euclidien $\mathbb{R}^2$ est l'un des espaces topologiques les plus utilisés en terme de plongement de graphes. Les graphes admettant un plongement dans $\mathbb{R}^2$ sont exactement les graphes planaires. Ce sont les graphes qui peuvent être plongés dans le plan sans croisement d'arêtes.

Un graphe plan est le plongement d'un graphe dans le plan sans croisement d'arête. Un graphe plan divise donc le plan en régions connexes appelées des *faces*. Les sommets de G sont représentés par des points de $\mathbb{R}^2$. Le contour d'une face d'un graphe plan G est un cycle non élémentaire de manière générale, et un cycle dans le cas où G est 2-connexe avec au moins 3 sommets.

On a le théorème d'Euler suivant :

Théorème 1.4 (*Théorème d'Euler*) *Si un graphe G à n sommets, m arêtes et f faces possède un plongement sur une surface orientable S_g de genre g , alors on a :*

$$n - m + f = 2 - 2g.$$

Donc, si G est planaire, alors on a $n - m + f = 2$. Or, on sait que dans ce plongement, chaque face possède au moins 3 arêtes et qu'une arête appartient à au plus deux faces. Donc, le nombre de faces de G est au plus $2m/3$. Ce qui veut dire que le nombre d'arêtes $m \leq 3n - 6$.

Chapitre 2

Décomposition arborescente des graphes planaires

2.1 Introduction

Les notions de *décomposition arborescente* et de *largeur arborescente* sont utilisées dans la résolution de beaucoup de problèmes algorithmiques. En effet, plusieurs de problèmes difficiles (souvent NP -complets) admettent des algorithmes polynomiaux, voir même linéaires pour les classes de graphes de largeur arborescente bornée [AHU72, Arn85, AP89, PT93, Bod87, Bod88, Bod90, Cou90, Cou96, MT92]. Comme exemple on peut citer deux problèmes assez connus : le problème d'isomorphisme de graphes et le problème de coloration de graphe [Bod90].

Les notions de décomposition arborescente et de largeur arborescente ont été introduites par Robertson et Seymour [RS86] dans leurs travaux sur les *mineurs* de graphe. L'ensemble de ces notions sera proprement défini dans la section suivante (cf. définition 2.1). Informellement, une décomposition arborescente d'un graphe G est un arbre avec des étiquettes sur les nœuds et chaque étiquette est un ensemble de sommets de G . La largeur d'une décomposition est le nombre maximum de sommets contenus dans une étiquette diminué de un. La largeur arborescente d'un graphe est le minimum sur toutes les largeurs de toutes les décompositions du graphe.

Pour tout graphe G , le problème de décider si la largeur arborescente de G est au plus k est NP -complet lorsque k fait partie de l'entrée [ACP87]. Cependant, il existe plusieurs classes de graphes admettant des algorithmes qui calculent la largeur arborescente en temps polynomial. Par exemple, on peut citer les graphes chordaux, les graphes de permutation [BKK95b] et les graphes d'intervalle (cf. [SSR94, Klö93, BKK95a, BM93, HM94] pour plus d'exemples). Par contre, ce problème est toujours ouvert pour les graphes planaires. Mais, il existe un algorithme polynomial d'approximation de la largeur arborescente des graphes planaires avec un facteur d'approximation de 1.5 [ST94].

Dans le cadre général, il existe un algorithme polynomial d'approximation de la largeur arborescente k de tout graphe G avec un facteur d'approximation de $O(\sqrt{\log k})$ [FHL08].

En outre, pour une constante k donnée, il existe un algorithme linéaire qui donne une décomposition arborescente de largeur au plus k si elle existe. En effet, il existe un algorithme qui en temps $2^{O(k)}n$ donne une décomposition arborescente de largeur

au plus $O(k)$ pour tout graphe G et toute constante k donnée [BGKH92]. Décider si $\text{tw}(G) \leq k$ est polynomial si G est de largeur arborescente bornée.

Il faut aussi noter que plusieurs résultats sur la largeur arborescente ont une connexion très forte avec la logique. Les travaux de Arnborg *et al* [ACPS93], Courcelle [Cou90] et Courcelle et Mosbah [CM93] montrent que tout problème exprimable en logique monadique du second ordre peut être résolu en temps linéaire dans les graphes de largeur arborescente bornée.

Cependant, bien que la notion de largeur arborescente soit très utile pour la résolution de certains problèmes, le diamètre des sacs reste aussi une notion importante pour la résolution de problèmes de distance. C'est dans ce sens que la notion de *longueur arborescente* a été introduite par Dourisboure et Gavaille [DG03, DG04a, DG04b]. Il est NP -complet de calculer la longueur arborescente [Lok07]. Dans ce même papier, il montre qu'il n'est pas possible d'approximation la longueur arborescente d'un graphe avec un facteur d'approximation plus petit que $3/2$. Cependant, il existe un algorithme d'approximation de la longueur arborescente, pour les graphes non valués avec un facteur d'approximation de 3 [DG04a]. De plus, dans [DG04a], Dourisboure et Gavaille ont montré qu'il n'existe pas de compromis entre la longueur et la largeur arborescente, c'est-à-dire, il existe des graphes pour lesquels il n'est pas possible de minimiser à la fois la largeur et la longueur arborescente. Il existe aussi plusieurs autres raisons qui de s'intéresser à ce nouveau paramètre. Par exemple, Dourisboure *et al.* les graphes de longueur arborescente bornés admettent un "Spanner" avec une constante additive près [DG04c]. Il a aussi été montré que beaucoup de classe de graphes de largeur arborescente non bornée sont de longueur arborescente bornée; c'est le cas des graphes chordaux, les graphes d'intervalle, les graphes de permutation, les graphes "split" et les graphes AT -free [DG07b]. Il y a aussi des travaux très récents reliant la longueur arborescente à d'autres graphes. Par exemple dans [CDE+08] un donne une liaison avec les graphes hyperboliques. Il a aussi été montré dans [UY09] que la longueur arborescente est équivalente à la longueur de branche.

Dans ce chapitre, nous nous intéressons à la décomposition arborescente des graphes de longueur arborescente au plus ℓ . Nous allons montrer que tout graphe planaire de longueur arborescente ℓ admet une décomposition arborescente dans laquelle chaque sac contient au plus 6 chemins de longueur au plus $3\ell/2$. De plus, chaque chemin est constitué d'au plus 2 plus courts chemins de longueur $\leq \ell$. Ce résultat permet de mettre en évidence une relation entre la longueur arborescente et la largeur arborescente d'un graphe. En d'autres termes, tout graphe planaire G de longueur arborescente ℓ admet une décomposition arborescente avec au plus 9ℓ sommets par sac. Nous montrons donc que $\text{tw}(G) = O(\ell)$. De même, on montre que G possède une décomposition dans laquelle chaque sac contient au plus 12 plus courts chemins de longueur $\leq \ell$.

Les grandes lignes de ce chapitre se présentent de la manière suivante.

Dans la section 2.2, nous allons donner une définition plus précise de ce qu'est la décomposition arborescente, la largeur et la longueur arborescente.

Dans la section 2.3, nous présentons deux résultats préliminaires sur la décomposition arborescentes des graphes de longueur arborescente ℓ . En effet, nous donnons des décompositions arborescentes des graphes de longueur arborescente ℓ dans les-

quelles on impose des conditions supplémentaires.

Enfin dans la section 2.4, nous donnerons le résultat principal de ce chapitre.

2.2 Décomposition arborescente

La décomposition arborescente a été introduite par Robertson et Seymour [RS86]. Pour plus de détails on peut se référer au livre de Downey et Fellow [DF99] et aux articles de synthèse de Bodlaender [Bod05, Bod06].

Définition 2.1 (Décomposition arborescente) *Pour tout graphe $G = (V, E)$, une décomposition arborescente (tree-decomposition en anglais) de G est un arbre $\mathcal{T}$ dont les nœuds appelés sacs (bags en anglais) sont des sous ensembles de sommets de G . Cet arbre respecte les trois conditions suivantes :*

- (i) *Tout sommet de G appartient à au moins un sac.*
- (ii) *Toute arête de G a ses deux extrémités contenues en même temps dans au moins un sac.*
- (iii) *Pour tout sommet u de G , l'ensemble des sacs contenant u induit un sous-arbre de $\mathcal{T}$.*

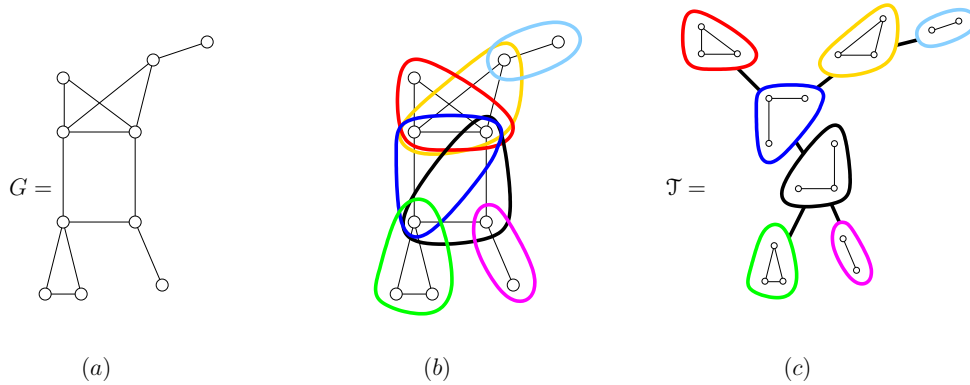


FIG. 2.1: (a) Un graphe G . (c) Une décomposition arborescente $\mathcal{T}$ de G . (b) Une matérialisation des sacs de $\mathcal{T}$ dans le graphe G .

On utilisera le terme *nœud* pour désigner les sommets des arbres et le terme *sac* pour désigner les nœuds d'une décomposition arborescente. Cette convention est commode lorsqu'on manipule à la fois un graphe et un arbre représentant sa structure.

La *largeur* d'une décomposition $\mathcal{T}$ est $\max_{X \in \mathcal{T}} |X| - 1$, et la *longueur* de $\mathcal{T}$ est $\max_{X \in \mathcal{T}} (\max_{u, v \in X} d_G(u, v))$.

La *largeur arborescente* et la *longueur arborescente* (respectivement *tree-width* et *tree-length* en anglais) d'un graphe G sont respectivement la plus petite largeur et la plus petite longueur parmi toutes les décompositions arborescentes de G . La

largeur arborescente et la longueur arborescente seront respectivement notées $\text{tw}(G)$ et $\text{tl}(G)$.

La définition de la longueur arborescente s'étend aux graphes valués, où $\text{dist}_G(u, v)$ représente la distance entre u et v dans G , c'est-à-dire le minimum de la somme des valuations des arêtes d'un chemin de u à v dans G . Cependant, sauf explicitement mentionnée, les graphes considérés sont non valués. Une décomposition arborescente peut ne comporter qu'un sac qui contient alors tous les sommets du graphe. Dans ce cas la largeur est égale au nombre de sommets du graphe moins 1.

Les arbres sont de largeur arborescente 1. Le cycle C_n à n sommets a une largeur arborescente 2 si $n \geq 3$; la longueur arborescente est $\lceil n/3 \rceil$ [DG04a]. Une clique à n sommets notée K_n , est de largeur arborescente $n - 1$. Pour une grille $p \times q$, la largeur arborescente est $\min\{p, q\}$ [Bod98]. Les graphes planaires extérieurs sont de largeur arborescente au plus 2.

Dans une décomposition arborescente, il est autorisé que des sacs soient inclus dans d'autres sacs. Mais toute décomposition arborescente $\mathcal{T}$ d'un graphe peut être transformée en une décomposition arborescente de ce graphe de même largeur sans qu'aucun sac ne soit inclus dans un autre. En effet, si un sac X de $\mathcal{T}$ est inclus dans un autre sac Y de $\mathcal{T}$, il suffit de contracter l'arête (X, Y) de $\mathcal{T}$ en fusionnant les deux sacs en un seul et en prenant X comme étant le sac associé au nœud résultant de cette fusion. Une décomposition arborescente dans laquelle aucun sac n'est inclus dans un autre est dite *réduite*. Une telle décomposition comprend au plus n sacs.

2.3 Décompositions arborescentes soumises à des conditions particulières

Il arrive parfois qu'on impose des conditions supplémentaires aux décompositions arborescentes pour certaines applications algorithmiques. En effet, les décompositions arborescentes réduites et les décompositions arborescentes de longueur minimale sont des décompositions soumises à des conditions particulières. Nous pouvons également citer les quelques résultats suivant :

Fraignaud et Nisse [PN06] ont montré que tout graphe connexe G admet une décomposition arborescente *connexe*, de largeur au plus $\text{tw}(G)$. Une décomposition arborescente $\mathcal{T}$ est *connexe* si pour toute arête e de $\mathcal{T}$, les sommets des sacs de chacun des deux sous-arbres de $\mathcal{T} \setminus \{e\}$ induisent un sous-graphe connexe de G .

Bodlaender et al. [BH98] et Courcelle *et al.* [CV03], ont montré que tout arbre à n nœuds possède une décomposition arborescente de largeur 2 dont le diamètre de l'arbre est au plus $3 \log n$. Ils ont aussi montré que tout graphe à n sommets possède une décomposition arborescente de largeur au plus $3k + 2$ et dont le diamètre de l'arbre est au plus $O(\log n)$; k est la largeur arborescente du graphe.

Dans [BE97], Bodlaender et Engelfriet ont montré que tout graphe G admet une décomposition arborescente de type *domino*¹ de largeur arborescente au plus une fonction de $\text{tw}(G)$ et du degré maximum de G . Il n'est pas possible de remplacer la borne par une fonction qui ne dépend que de $\text{tw}(G)$.

¹Une décomposition est de type *domino* si aucun sommet n'appartient à plus de 2 sacs.

Dans cette partie, nous allons donner deux algorithmes qui, étant donné un graphe G , produisent chacun une décomposition arborescente de G , respectant certaines conditions. Ces algorithmes supposent l'existence d'une décomposition arborescente $\mathcal{T}_0$ de G de longueur arborescente ℓ . L'objectif est de modifier $\mathcal{T}_0$ en un nouvel arbre de décomposition arborescente $\mathcal{T}$ de longueur $O(\ell)$, respectant certaines conditions.

2.3.1 Décomposition arborescente où chaque sac possède une partie privée

Afin de démontrer notre résultat principal, nous utiliserons des décompositions arborescentes où chaque sac X possède une partie unique pour X (partie privée). Notamment, les propriétés de cette décomposition sont utilisées pour la propriété 2.8.

Définition 2.2 (Partie privée) *Une partie privée d'un sac X d'une décomposition arborescente est un sous-ensemble de sommets qui appartient exclusivement à X .*

Dans cette partie, nous allons montrer que toute décomposition arborescente $\mathcal{T}_0$ de longueur ℓ d'un graphe G peut être transformée en une décomposition arborescente $\mathcal{T}$ de G dans laquelle chaque sac X de $\mathcal{T}$ admet une partie privée avec un sommet qui est à distance au plus ℓ de tout sommet $y \in X$. De plus, on peut remarquer que dans ce cas, la longueur de $\mathcal{T}$ est au plus 2ℓ . En effet, pour tout sac X de $\mathcal{T}$ avec x un sommet privé de X , et pour tout couple de sommets $(y, z) \in X$, on a $\text{dist}_G(y, z) \leq \text{dist}_G(y, x) + \text{dist}_G(x, z) \leq 2\ell$.

Par exemple sur la figure 2.2, on a en (b) une décomposition $\mathcal{T}$ du graphe G représenté en (a). Chaque sac de $\mathcal{T}$ a une partie privée. Les sommets privés de chaque sac sont colorés en rouge. Par exemple, le sommet privé du sac bleu sont le sommet i tandis que les sommets privés du sac noir sont les sommets d et e .

Pour tout graphe G , il existe une décomposition arborescente $\mathcal{T}$ dans laquelle chaque sac admet une partie privée. En effet, il suffit de considérer la décomposition arborescente constituée d'un seul sac.

Par ailleurs, on peut remarquer que si $\mathcal{T}$ est une décomposition arborescente d'un graphe G dans laquelle chaque sac admet une partie privée, alors tout sac contient le voisinage de sa partie privée. En effet, si un sommet x d'une partie privée d'un sac X avait un voisin y dans G , alors par la condition (ii) de la définition 2.1, il existe un sac contenant $\{x, y\}$. Ce sac ne peut qu'être X .

Nous allons montrer le résultat suivant qui est utile pour la suite :

Lemme 2.3 *Soit G un graphe valué ayant une décomposition arborescente de longueur ℓ . Alors G possède une décomposition arborescente $\mathcal{T}$ où chaque sac X de $\mathcal{T}$ possède une partie privée avec un sommet x telle que $\text{dist}_G(x, y) \leq \ell$ pour tout $y \in X$.*

Démonstration : On considère une décomposition arborescente $\mathcal{T}$ de G de longueur $\leq \ell$. On suppose que $\mathcal{T}$ est réduite, cela ne change pas sa longueur. On supposera aussi que $\mathcal{T}$ est enracinée en un sac R . Pour chaque sac X , on note $\text{père}(X)$ le père de X dans $\mathcal{T}$, avec la convention $\text{père}(R) = \emptyset$.

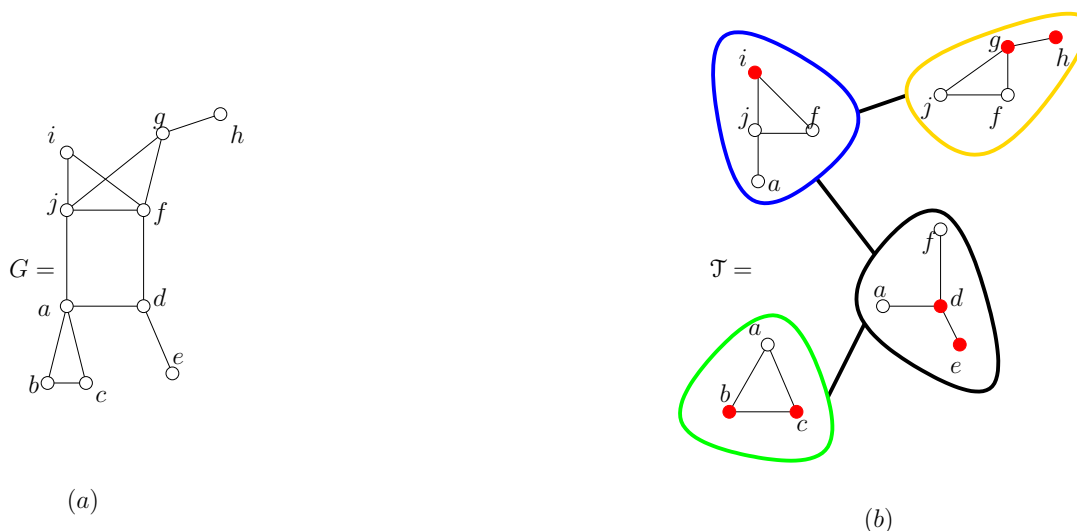


FIG. 2.2: (a) Un graphe G . (b) Une décomposition arborescente $\mathcal{T}$ avec parties privées de G .

On va modifier $\mathcal{T}$ grâce à une procédure appelée **CONTRACT**($\mathcal{T}, X$) qui modifie le sous-arbre $\mathcal{T}_X$ de $\mathcal{T}$ (sous-arbre de $\mathcal{T}$ de racine X) et qui construit pour chaque sac de ce nouveau sous-arbre une partie privée. Rappelons que contracter une arête (X, Y) de $\mathcal{T}$ revient à remplacer le sac X par $X = X \cup Y$ et de relier X à tout sac Z adjacent $X \cup Y$ dans $\mathcal{T}$.

Initialement on lance **CONTRACT**($\mathcal{T}, R$)

CONTRACT($\mathcal{T}, X$) :

1. Choisir $x \in X \setminus \text{père}(X)$
2. Contracter les arêtes des chemins de X vers tout descendants Y de X dans $\mathcal{T}$ tel que $x \in Y$. Mettre à jour $\mathcal{T}$ et X .
3. Pour chaque fils Z de X dans ce nouvel arbre $\mathcal{T}$, appeler **CONTRACT**($\mathcal{T}, Z$)

Montrons que la procédure **CONTRACT** termine en temps polynomial en n . Pour cela, nous montrons que chacune des 3 étapes de l'algorithme termine et que le temps additionnel sur toutes les étapes est polynomial en n .

D'abord, montrons que l'étape 1 termine, en temps linéaire en n . Pour cela, il suffit de montrer que le choix de x se fait normalement quelque soit le sac X donné. En effet, ce qui pourrait empêcher cette étape de se faire normalement c'est qu'il existe une entrée X pour laquelle x n'existe pas. Dans ce cas, vu que $X \neq \emptyset$, alors on aurait $X \subseteq \text{père}(X)$. On a donc une contradiction avec le fait que $\mathcal{T}$ est une décomposition réduite. D'où x existe et donc l'étape 1 se termine en temps linéaire en n .

Ensuite, montrons que l'étape 2 termine en temps polynomial quelque soit les entrées $\mathcal{T}$ et X . En effet, on sait que l'étape 2 consiste à vérifier pour chaque des-

endant Y de X si $x \in Y$ et le cas échéant, contracter les arêtes du chemin de X à Y . On sait que $|V(\mathcal{T})| \leq n$ car $\mathcal{T}$ est réduit. De plus, un sac ne peut contenir plus de n sommets. Donc, en supposant que la contraction des arêtes sur un chemin se fait en temps constant, l'étape 2 termine en temps polynomial en n .

Enfin, montrons que l'étape 3 termine temps polynomial en n quelque soit les entrées $\mathcal{T}$ et X . En effet, l'étape 3 fait un appel récursif de la procédure **CONTRACT** sur tous les fils du sac X (s'il y en a). Donc étant donné que les étapes 1 et 2 terminent quelque soit l'entrée $X, \mathcal{T}$, et que X ne peut avoir plus de $|V(\mathcal{T})|$ fils, alors l'étape 3 termine en $O(|V(\mathcal{T})|n)$. D'où, la procédure termine en temps polynomial en n .

Montrons maintenant que la procédure **CONTRACT** produit bien un arbre $\mathcal{T}$ de décomposition arborescente de G dans lequel chaque sac admet une partie privée.

En effet, on sait que l'étape 1 consiste à choisir un sommet $x \notin \text{père}(X)$. L'étape 2 consiste à contracter les arêtes des chemins de X à tous ses descendants Y contenant x et à mettre à jour $\mathcal{T}$ et X . Il est certain qu'après l'étape 2 le sac X devient le seul sac de $\mathcal{T}$ à contenir le sommet x . Donc le sac X a au moins un sommet privé qui est x . Par ailleurs, on appelle la procédure **CONTRACT** initialement sur R , la racine de $\mathcal{T}$. Donc elle construit en premier un sommet privé pour R . De plus, vu que l'étape 3 consiste à faire des appel récursifs sur les fils du sac en cours, alors tous les sacs de $\mathcal{T}$ seront traités un à un et donc à la sortie, chaque sac aura une partie privée.

Montrons que que l'arbre $\mathcal{T}$ produit est un arbre de décomposition arborescente de G . Pour cela, il suffit de montrer que la tâche effectuée par l'étape 2 ne modifie pas les propriétés de $\mathcal{T}$ car les étapes 1 et 3 n'ont pas d'action directe sur les sacs.

En effet, la tâche effectuée en 2 consiste à contracter des arêtes de l'arbre de décomposition $\mathcal{T}$ donné en paramètre. Or on sait que la contraction d'arêtes dans un arbre de décomposition ne modifie pas les propriétés de décomposition. Donc l'arbre produit après l'étape 2 est bien une décomposition arborescente de G .

Nous avons ainsi montré que pour tout graphe G et tout arbre $\mathcal{T}$ de décomposition arborescente de G enraciné en un sac R et de longueur ℓ , **CONTRACT**($\mathcal{T}, R$) produit un arbre de décomposition arborescente de G dans lequel chaque sac admet une partie privée.

Montrons que si $\mathcal{T}$ est un arbre de décomposition de G produit par la procédure **CONTRACT**, alors chaque sac X de $\mathcal{T}$ possède un sommet x tel que $\forall y \in X$, $\text{dist}_G(x, y) \leq \ell$.

Soit $\mathcal{T}_0$ l'arbre de décomposition avant l'application de la procédure **CONTRACT**. Pour tout sac X de $\mathcal{T}$, notons X_0 le sac correspondant à X dans $\mathcal{T}_0$. Le sac X est produit par **CONTRACT**($\mathcal{T}, X_0$). Soit $r(X)$ le sommet x choisi dans l'étape 1 de **CONTRACT**($\mathcal{T}, X_0$). Calculons maintenant pour tout sommet $y \in X$, la distance $\text{dist}_G(r(X), y)$.

Si $X = X_0$, alors X_0 n'est pas modifié. Donc X_0 est le seul sac de $\mathcal{T}_0$ qui contient $r(X)$. Le sommet $r(X)$ est à distance au plus ℓ de tout sommet y de X_0 car $\mathcal{T}_0$ est de longueur au plus ℓ . Donc $\text{dist}_G(r(X), y) \leq \ell$.

Sinon $X \neq X_0$. Alors il existe un ensemble de sacs $Z_{01}, \dots, Z_{0k}$ de $\mathcal{T}_0$ contenant

$r(X)$ tel que $X = X_0 \cup Z_{01} \cup \dots \cup Z_{0k}$. Vu que $\mathcal{T}_0$ est de longueur ℓ , alors pour tout $y \in Z_{0i \in 1 \leq i \leq k}$ on a $d_G(r(X), y) \leq \ell$. Donc pour tout sommet $y \in X$, on a $d_G(r(X), y) \leq \ell$. Ainsi, $\mathcal{T}$ est un arbre de décomposition de G dont chaque sac X admet une partie privée avec un sommet $r(X)$ qui est à distance au plus ℓ de tout sommet $y \in X$. Ce qui termine la preuve du lemme 2.3. $\square$

2.3.2 Décomposition arborée

Nous pouvons remarquer que la décomposition donnée par le lemme 2.3 ne garantit pas que chaque sac X possède un arbre couvrant X exactement. Cette propriété sera nécessaire pour démontrer notre résultat principal. Ainsi, nous avons besoin des définitions suivantes :

Soit G un graphe et T un sous-arbre de G . On suppose T enraciné en un sommet r de G . On dit que T est k -monotone si pour tout $v \in T$, le chemin dans T entre v et r est composé d'au plus k plus courts chemins dans G .

Définition 2.4 (Décomposition arborée) *Une décomposition arborée d'un graphe G est une paire $(\mathcal{T}, \mathcal{S})$ telle que $\mathcal{T}$ est une décomposition arborescente de G et $\mathcal{S} = (T_X)_{X \in \mathcal{T}}$ est une famille d'arbres de G telle que tout sac $X \in \mathcal{T}$ possède une partie privée et un arbre $T_X \in \mathcal{S}$ tel que $V(T_X) = X$ et enraciné en un sommet de la partie privée de X .*

De plus, si une décomposition arborée $(\mathcal{T}, \mathcal{S})$ de G est telle que tout arbre de $\mathcal{S}$ est k -monotone et de profondeur $\leq h$, alors on dit que $(\mathcal{T}, \mathcal{S})$ est k -monotone et de profondeur h .

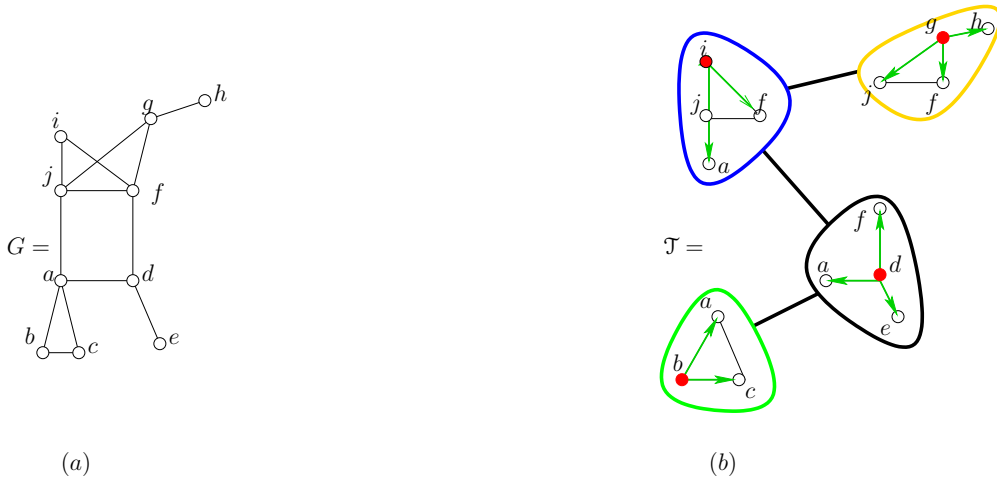


FIG. 2.3: (a) Un graphe G . (b) Une décomposition arborée $(\mathcal{T}, \mathcal{S})$ de G 1-monotone de profondeur deux.

Par exemple, la figure 2.3 (b) présente une décomposition arborée $(\mathcal{T}, \mathcal{S})$ du graphe G représenté en (a). Dans cet exemple la racine de chaque arbre pour chaque

sac est un sommet de la partie privée du sac. Sur la figure, chaque arbre est représenté par des arêtes vertes orientées, partant de la racine représentée par un sommet coloré en rouge.

Il est clair que tout graphe connexe admet une décomposition arborée. En effet, il suffit de considérer la décomposition arborescente composée d'un sac $X = V(G)$, et un arbre couvrant G .

Nous allons montrer le résultat suivant :

Théorème 2.5 *Soit G un graphe valué ayant une décomposition arborescente de longueur ℓ . Alors on peut construire en temps polynomial une décomposition arborée $(\mathcal{T}, \mathcal{S})$ de G qui est 2-monotone et de profondeur $3\ell/2$.*

On laisse comme question ouverte de savoir si dans l'énoncé du théorème 2.5 on peut remplacer "2-monotone" par "1-monotone".

Toute la suite de cette partie sera consacrée à la preuve du théorème 2.5.

Soit T une décomposition arborescente de G de longueur ℓ . Par le lemme 2.3, on peut supposer $\mathcal{T}_0$, un arbre de décomposition obtenu à partir de T et tel que chaque sac X de $\mathcal{T}_0$ possède une partie privée avec un sommet, noté $r(X)$, tel que $d_G(r(X), y) \leq \ell$ pour tout $y \in X$.

On va transformer $\mathcal{T}_0$ en une décomposition arborée $\mathcal{T}$ grâce à la procédure **ARBORER**($\mathcal{T}_0$). Au cours de la transformation de $\mathcal{T}_0$, il va apparaître un certain nombre de sacs qui ne respecteront pas les conditions de la définition 2.1. Ainsi, nous aurons à les corriger tout au long de la transformation pour qu'à la sortie de la procédure l'arbre $\mathcal{T}$ obtenu vérifie les trois conditions de la définition. Pour cela, nous avons besoin d'énoncer les spécifications suivantes :

Pour tout arbre T de décomposition (T n'est pas forcément une décomposition arborescente), soit $\mathcal{B}(T)$ l'ensemble des paires de sacs $\{X, Y\}$ pour lesquelles il existe au moins un sac $Z \in T[X, Y]$ tel que $X \cap Y \not\subseteq Z$. $T[X, Y]$ est le chemin dans $\mathcal{T}$ entre X et Y . Les paires de sacs contenues dans $\mathcal{B}(T)$ sont appelées des mauvaises paires. Il est clair que si T est une décomposition arborescente, alors $\mathcal{B}(T) = \emptyset$, car l'existence d'une mauvaise paire dans T fait que T ne vérifie pas la condition (iii) de la définition 2.1.

Soit $\mathcal{M}(T)$ le sous-ensemble de toutes les paires $\{X, Y\}$ de $\mathcal{B}(T)$ telles que $d_T(X, Y)$ est maximum.

ARBORER($\mathcal{T}_0$) :

1. Calculer $\mathcal{T}$ un arbre de décomposition à partir de $\mathcal{T}_0$ en modifiant les sacs de la façon suivante : calculer pour tout sac X_0 de $\mathcal{T}_0$ un arbre T_{X_0} couvrant X_0 de plus courts chemins dans G , enraciné en $r(X_0)$, et poser $X_0 = V(T_{X_0})$. Calculer $\mathcal{M}(\mathcal{T})$.
2. Tant que $\mathcal{M}(\mathcal{T}) \neq \emptyset$, prendre $\{X, Y\} \in \mathcal{M}(\mathcal{T})$ et faire :
 - 2.1. Pour tout sac $Z \in \mathcal{T}[X, Y]$, faire $Z = Z \cup (X \cap Y)$.
 - 2.2. Mettre à jour $\mathcal{M}(\mathcal{T})$.
3. Pour tout sac X de $\mathcal{T}$, calculer T_X un arbre couvrant $G[X]$, enraciné en $r(X)$ et de plus courts chemins dans $G[X]$.

Soit $\mathcal{T}_i$ l'arbre produit après l'étape i et pour tout sac $X_0 \in \mathcal{T}_0$, soit X_i le sac correspondant à X_0 dans l'arbre $\mathcal{T}_i$, avec $1 \leq i \leq 3$. On note $\mathbb{S} = (T_X), X \in V(\mathcal{T}_3)$ la famille des arbres construit à l'étape 3.

Montrons que si la procédure termine, alors l'arbre $\mathcal{T}_3$ produit est un arbre de décomposition arborescente de G . Pour cela, nous allons montrer que l'arbre $\mathcal{T}_3$ respecte bien les trois conditions de la définition 2.1.

Nous pouvons remarquer que l'étape 3 consiste à construire pour chaque sac $X \in \mathcal{T}_2$ un arbre T_X couvrant $G[X]$ de plus courts chemins. Donc l'étape 3 ne modifie pas les propriétés de $\mathcal{T}_2$. Donc pour montrer que l'arbre $\mathcal{T}_3$ respecte bien les conditions, il suffit de montrer que $\mathcal{T}_2$ respecte les trois conditions de la définition 2.1.

D'abord, montrons que $\mathcal{T}_2$ respecte bien les conditions (i),(ii) de la définition 2.1. En effet, si $\mathcal{T}_2$ ne respecte pas la condition (i) ou (ii), alors il existe au moins un sommet x de G qui n'est dans aucun sac de $\mathcal{T}_2$. Vu que $\mathcal{T}_0$ est une décomposition arborescente de G , il existe au moins un sac X_0 de $\mathcal{T}_0$ qui contient x . Vu que $x \notin X_2$ et $x \in X_0$, alors la transformation de X_0 en X_2 par la procédure a supprimé le sommet x de X_0 pour former le sac X_2 . Or on sait que ceci est impossible car aucune des étapes de la procédure n'a pour tâche d'enlever des éléments à un sac. Donc $\mathcal{T}_2$ respecte bien les conditions (i) et (ii) de la définition 2.1.

Enfin, montrons que $\mathcal{T}_2$ respecte bien la condition (iii) de la définition 2.1. En effet, supposons que $\mathcal{T}_2$ ne respecte pas la condition (iii). Alors il existe au moins deux sacs, X_2 et Y_2 tels qu'il existe un sac $Z_2 \in \mathcal{T}[X_2, Y_2]$ et $X_2 \cap Y_2 \notin Z_2$. En d'autres termes, dans $\mathcal{T}_2$ il y a au moins une mauvaise paire de sacs $\{X_2, Y_2\}$. Donc $\mathcal{M}(\mathcal{T}_2) \neq \emptyset$; ce qui contredit la terminaison de la procédure car la condition d'arrêt de l'étape 2 est que $\mathcal{M}(\mathcal{T}_2)$ soit vide. L'arbre $\mathcal{T}_2$ respecte donc la condition (iii). Par conséquent, nous avons montré que l'arbre produit par la procédure **ARBORER** est bien un arbre de décomposition arborescente.

Montrons maintenant que l'arbre $\mathcal{T}_3$ produit est tel que chaque sac X_3 de $\mathcal{T}_3$ possède un arbre couvrant X_3 de plus courts chemins dans $G[X_3]$. Il est clair que pour tout sac X_2 de $\mathcal{T}_2$ tel que $G[X_2]$ est connexe, l'étape 3 construit un arbre couvrant X_2 de plus courts chemins dans $G[X_2]$ en temps polynomial en $|X_2|$. Supposons qu'il existe un sac X_3 de $\mathcal{T}_3$ ne possédant pas d'arbre couvrant X_3 . Alors le sous graphe $G[X_2]$ est non connexe. Il existe au moins un sommet $x \in X_2$ non accessible à $r(X_1)$ dans $G[X_2]$. On a donc $x \notin X_1$ car, $G[X_1]$ est connexe grâce à l'étape 1. En

d'autres termes, le sommet x est ajouté au sac X_1 pour former le sac X_2 pendant l'étape 2. Par définition de l'étape 2 de la procédure, il existait à cet instant du processus au moins une paire $\{W_1, Y_1\}$ de sacs de $\mathcal{T}_1$ telles que $x \in W_1 \cap Y_1$ et $x \notin X_1$. Soit $\{W_1, Y_1\}$ la mauvaise paire de $\mathcal{T}_1$ telle que $x \in W_0$ et $\text{dist}_{\mathcal{T}_1}(W_1, Y_1)$ maximale. Il existe un sac Y_0 de $\mathcal{T}_0$ et un sommet $v \in Y_0$ tel que $x \in T_{Y_0}[r(Y_0), v]$ car, l'étape 1 consiste à construire pour chaque sac X_0 de $\mathcal{T}_0$ un arbre T_{X_0} de plus courts chemins dans G couvrant X_0 et en considérant le sac X_1 comme étant l'ensemble des sommets de T_{X_0} . C'est pendant la construction de l'arbre T_{Y_0} que le sommet x s'est retrouvé dans T_{Y_0} et donc dans Y_1 . On a donc $x \in T_{Y_0}[r(Y_0), v]$. On a le chemin $P = T_{Y_0}[r(Y_0), v] \subseteq Y_1$ par définition. De plus étant donné que $x \notin X_1$, alors à cet instant la paire $\{W_1, Y_1\}$ est une mauvaise paire. Cette mauvaise paire sera corrigée par l'étape 2 en posant $X_2 = X_1 \cup \{W_1 \cap Y_1\}$. En particulier le sommet x est dans X_2 .

Par ailleurs, du fait que $X_0 \in \mathcal{T}_0[W_0, Y_0]$, alors on a X_0 qui est un $(r(Y_0), x)$ -séparateur² de G car $\mathcal{T}_0$ est un arbre de décomposition de G . Donc, le chemin P précédemment défini contient au moins un sommet de X_0 . Si $|P \cap X_0| = 1$, alors en posant $P \cap X_0 = \{z\}$, on a la sous chaîne de P entre $r(Y_1)$ et z union la sous chaîne de P entre z et v qui est un chemin entre $r(Y_1)$ et v plus court que P . Ce qui est une contradiction avec le fait que P est un plus court chemin dans G . Donc $|P \cap X_0| \geq 2$. Soit p_1 la sous chaîne de P entre X_0 et x , et soit p_2 la sous chaîne de P disjointe de p_1 entre x et X_0 . On a $p_1 \cup p_2 \subseteq Y_1$. Vu que X_2 est non connexe, il existe au moins un sommet $z_1 \in p_1$ et un sommet $z_2 \in p_2$ tels que $z_1, z_2 \notin X_2$. En effet, si on avait $p_1 \subseteq X_2$ ou $p_2 \subseteq X_2$, alors on aurait dans X_2 un chemin reliant $r(X_2)$ à x ; ce qui contredit l'hypothèse. Soit Z_0 le sac de $\mathcal{T}_0$ contenant z_1 . On a $X_0 \in \mathcal{T}_0[Y_0, Z_0]$. Vu que $z_1 \in Z_2$ et $z_1 \in Y_2$, alors on a $\{Y_2, Z_2\}$ qui est une mauvaise paire de $\mathcal{T}_2$. Ce qui contredit la terminaison de la procédure car, la condition d'arrêt de l'étape 2 est que dans $\mathcal{T}_2$, il n'y a plus de mauvaise paire. Donc, on a $p_1 \subseteq X_2$ et $p_2 \subseteq X_2$, c'est-à-dire $G[X_2]$ est connexe. Nous avons donc montré que tout sac X_2 de $\mathcal{T}_2$ est tel que $G[X_2]$ est connexe et donc X_2 admet un arbre couvrant de plus courts chemins dans $G[X_2]$.

De plus on a $\text{dist}_{T_X}(r(X), x) \leq \ell + \min\{|p_1|, |p_2|\}$. Or on a $|P| \leq \ell$ où $|P|$ est le nombre d'arêtes de P (plutôt le coût du chemin P). Donc on a soit $|p_1| \leq \ell/2$, soit $|p_2| \leq \ell/2$. Car $|P| \geq |p_1| + |p_2|$. On a donc $\text{dist}_{T_X}(r(X), x) \leq 3\ell/2$. Par conséquent, la profondeur de $\mathcal{T}_3$ est $\leq 3\ell/2$.

Vérifions maintenant que $T_{X_0}[r(X_0), x]$ est constitué de deux plus courts chemins dans G . En effet, on a $T_{X_0}[r(X_0), x] = T_{X_0}[r(X_0), z_1] \cup T_{X_0}[z_1, x]$. Or $z_1 \in X_0$, donc $T_{X_0}[r(X_0), z_1]$ est un plus court chemin dans G par définition de l'étape 1. Par ailleurs, on a $T_{X_0}[z_1, x]$ qui est une sous chaîne de P . Alors $T_{X_0}[z_1, x]$ est aussi un plus court chemin dans G car P est un plus court chemin dans G . D'où $T_{X_0}[r(X_0), x]$ est constitué par deux plus courts chemins dans G .

Or, tout sommet de $x_0 \in X_0$ est accessible depuis $r(X_0)$ par de plus courts chemins dans G par définition de l'étape 1. Donc, nous avons montré que l'arbre T_{X_0} est 2-monotone.

²On rappelle qu'un ensemble F de sommets de G est un (z, t) -séparateur de G si z et t sont dans deux composante connexes différentes de $G \setminus F$ (voir Chapitre 1).

Nous avons donc montré que si la procédure **ARBORER** termine, elle construit une décomposition arborée de G , 2-monotone et de profondeur au plus $3\ell/2$. Montrons que la procédure **ARBORER**($\mathcal{T}$) termine en temps polynomial pour tout $\mathcal{T}$. Pour cela, il suffit de montrer que chacune des quatre étapes de la procédure termine en temps polynomial en n .

D'abord, montrons que l'étape 1 termine. Vu que G est connexe, il est évident que le calcul de l'ensemble des arbres couvrants chaque sac de $\mathcal{T}_0$ se fait en temps polynomial en n . En effet, pour tout sac X_0 de $\mathcal{T}_0$, le calcul d'un arbre couvrant X_0 se fait en temps au plus $O(n)$. De plus, on a $|\mathcal{T}_0| \leq n$, car $\mathcal{T}_0$ est réduit. Donc le calcul d'arbres couvrants pour tous les sacs de $\mathcal{T}_0$ se fait en temps $O(n^2)$.

Grâce à l'étape 1, nous avons modifié l'arbre $\mathcal{T}_0$ en un nouvel arbre $\mathcal{T}_1$ dans lequel chaque sac X_1 possède un arbre T_{X_0} couvrant exactement X_0 de plus courts chemins dans G . Pour chaque sac X_0 de $\mathcal{T}_0$, l'arbre T_{X_0} peut contenir des sommets qui ne sont pas dans X_0 . De ce fait, en posant $X_1 = T_{X_0}$, on autorise qu'un sommet $z \notin X_0$, appartenant à un sac Z_0 de $\mathcal{T}_0$ soit un sommet de X_1 et Z_1 dans $\mathcal{T}_1$. Mais, du fait que nous n'assurons pas que z soit dans tout sac $W \in \mathcal{T}_1[X_1, Z_1]$, nous autorisons que $\mathcal{T}_1$ ne respecte pas la condition (iii) de la définition 2.1. Si $\mathcal{T}_1$ ne vérifie pas (iii), alors il existe une mauvaise paire dans $\mathcal{B}(\mathcal{T})$. Le calcul de l'ensemble $\mathcal{B}(\mathcal{T}_1)$ consiste à calculer l'ensemble des mauvaises paires de $\mathcal{T}_1$. Or on sait que le nombre de couples de sacs est en $O(n^2)$. Le calcul de $\mathcal{B}(\mathcal{T}_1)$ et $\mathcal{M}(\mathcal{T}_1)$ se fait donc en temps polynomial en n . Nous avons donc montré que l'étape 1 termine en temps polynomial en n .

Ensuite, nous montrons ici que l'étape 2 termine en temps polynomial. On sait que l'étape 2 continue tant que $\mathcal{B}(\mathcal{T}_1) \neq \emptyset$. Alors pour montrer la terminaison de l'étape 2, nous allons suivre pas à pas l'évolution de l'ensemble $\mathcal{B}$ lors de l'exécution. Nous allons en fait montrer que $|\mathcal{B}(\mathcal{T}_1)|$ converge vers zéro. En effet, s'il n'y a pas de mauvaise paire de sacs dans $\mathcal{T}_1$, alors $\mathcal{B}(\mathcal{T}_1)$ est vide et dans ce cas l'algorithme ne rentre pas dans la boucle et l'étape 2 termine en temps constant. Sinon, il existe au moins une mauvaise paire dans $\mathcal{B}(\mathcal{T}_1)$. Dans ce cas, on a $\mathcal{M}(\mathcal{T}_1)$ qui est non vide. Soit $\{X_1, Y_1\}$ une paire de $\mathcal{M}(\mathcal{T}_1)$ telle que $\text{dist}_{\mathcal{T}_1}(X_1, Y_1) = k$. Par définition de $\mathcal{M}(\mathcal{T}_1)$, toute mauvaise paire $\{X'_1, Y'_1\}$ de $\mathcal{B}(\mathcal{T}_1) \setminus \mathcal{M}(\mathcal{T}_1)$ est telle que $\text{dist}_{\mathcal{T}_1}(X'_1, Y'_1) < k$.

L'étape 2.1 consiste à corriger la mauvaise paire $\{X_1, Y_1\}$ en copiant l'ensemble $X_1 \cap Y_1$ dans tout sac $Z_1 \in \mathcal{T}_1[X_1, Y_1]$. Il est clair qu'après cette opération, le couple de sacs $\{X_1, Y_1\}$ n'est plus une mauvaise paire. Par conséquent la mise à jour de $\mathcal{B}(\mathcal{T}_1)$ effectuée dans l'étape 2.2 enlève le couple $\{X_1, Y_1\}$ de $\mathcal{B}(\mathcal{T}_1)$ et donc de $\mathcal{M}(\mathcal{T}_1)$. D'où, en montrant que l'étape 2.1 n'a pas créé de nouvelles mauvaises paires de sacs $\{W_1, R_1\}$ dans $\mathcal{T}_1$ tel que $\text{dist}_{\mathcal{T}_1}(W_1, R_1) \geq k$, alors on montre que l'ensemble des mauvaises paires de $\mathcal{M}(\mathcal{T}_1)$ diminue strictement. On remarque que $\mathcal{B}(\mathcal{T}_1)$ peut augmenter alors que $\mathcal{M}(\mathcal{T}_1)$ diminue (le nombre de mauvaises paires à distance strictement inférieure à k peut soudainement augmenter). Cependant, $|\mathcal{M}(\mathcal{T}_1)| = 0$ implique $\mathcal{B}(\mathcal{T}_1) = \emptyset$. Donc $\mathcal{B}(\mathcal{T}_1)$ converge vers $\emptyset$. Montrons par l'absurde que l'étape 2.1 ne crée aucune nouvelle mauvaise paire de sacs dont la distance dans $\mathcal{T}$ est supérieure ou égale à k . Rappelons que la tâche affectée à l'étape 2.1 est pour toute mauvaise paire $\{X_1, Y_1\}$ de $\mathcal{M}(\mathcal{T}_1)$, de rajouter l'ensemble des sommets de $X_1 \cap Y_1$ dans tout sac $Z \in \mathcal{T}_1[X_1, Y_1]$. Donc, il est clair que l'étape 2.1 ne supprime en aucun cas les sommets d'un sac. Soit $\{X_1, Y_1\}$ une mauvaise paire de $\mathcal{M}(\mathcal{T}_1)$ telle que $\text{dist}_{\mathcal{T}_1}(X_1, Y_1) = k$.

Supposons que l'exécution de l'étape 2.1 sur le couple $\{X_1, Y_1\}$ crée une nouvelle mauvaise paire $\{Z, W\}$ de sacs tels que $\text{dist}_{\mathcal{T}_1}(Z, W) \geq k$. Vu que l'étape 2.1 ne modifie que des sacs qui se trouvent sur le chemin $\mathcal{T}_1[X_1, Y_1]$, alors au moins un des deux sacs Z et W se trouve sur le chemin $\mathcal{T}_1[X_1, Y_1]$. En effet, supposons qu'aucun des deux sacs ne se trouve sur le chemin $\mathcal{T}_1[X_1, Y_1]$. Alors le sac $R \in \mathcal{T}_1[Z, W]$ qui est tel que $Z \cap W \not\subseteq R$ est aussi sur le chemin $\mathcal{T}_1[X_1, Y_1]$ car si tel n'était pas le cas, alors ni R ni Z ni W ne serait pas modifié par l'étape 2.1. Donc la paire $\{W, Z\}$ était une mauvaise paire bien avant l'exécution de l'étape 2.1 sur $\{X_1, Y_1\}$; ce qui est une contradiction avec l'hypothèse. Donc, on a $R \in \mathcal{T}_1[X_1, Y_1]$ et c'est en ajoutant $X_1 \cap Y_1$ à R que la paire $\{W, Z\}$ est devenue une mauvaise paire. Ce qui est impossible car la paire $\{W, Z\}$ ne peut être rendue mauvaise qu'en enlevant des sommets du sac R , ce qui est impossible à l'étape 2.1. Donc la paire $\{W, Z\}$ ne peut pas être rendue mauvaise en agissant uniquement sur le sac R . D'où, au moins un des deux sacs Z, W est sur le chemin entre X et Y . Si tous les deux sacs Z et W sont tous sur le chemin $\mathcal{T}_1[X_1, Y_1]$, alors le sac R est aussi sur le chemin $\mathcal{T}_1[X_1, Y_1]$. Soit t un sommet de $W \cap Z$ tel que $t \notin R$. Vu que la tâche affectée à l'étape 2.1 consiste uniquement à copier $X_1 \cap Y_1$ dans tous les sacs sur le chemin $\mathcal{T}_1[X_1, Y_1]$, alors on a $X_1 \cap Y_1 \subset Z$, $X_1 \cap Y_1 \subset W$ et $X_1 \cap Y_1 \subset R$. Donc, $t \notin X_1 \cap Y_1$ car sinon t serait dans R . D'où on a $t \in Z \setminus \{X_1 \cap Y_1\}$, $t \in W \setminus \{X_1 \cap Y_1\}$ et $t \notin R \setminus \{X_1 \cap Y_1\}$. De plus, on sait que $Z \setminus \{X_1 \cap Y_1\}$, $W \setminus \{X_1 \cap Y_1\}$ et $R \setminus \{X_1 \cap Y_1\}$ représente respectivement les sacs Z_1 , W_1 et R_1 avant l'exécution de l'étape 2.1 sur $\{X_1, Y_1\}$. D'où, $\{W_1, Z_1\}$ était bien une mauvaise paire avant l'exécution de l'étape 2.1 sur la paire $\{X_1, Y_1\}$; ce qui contredit l'hypothèse. Donc au plus un des deux sacs Z et W est sur le chemin $\mathcal{T}_1[X_1, Y_1]$. On peut supposer sans perdre de généralité que le sac Z est sur le chemin $\mathcal{T}_1[X_1, Y_1]$. D'où, R n'est pas sur le chemin $\mathcal{T}_1[X_1, Y_1]$ car sinon l'étape 2.1 lui aurait ajouté les mêmes sommets qu'il a ajouté à Z et dans ce cas on aurait $W \cap Z \subseteq R$. Soit z un sommet de W qui est ajouté à Z en exécutant l'étape 2.1 sur $\{X_1, Y_1\}$. Donc, on a $z \in X_1 \cap Y_1$. Or par définition, l'exécution de l'étape 2.1 sur $\{X_1, Y_1\}$ ne modifie ni X_1 ni Y_1 . Donc les paires $\{X_1, W\}$ et $\{Y_1, W\}$ étaient deux mauvaises paires avant l'application de l'étape 2.1 sur le couple $\{X_1, Y_1\}$. D'où, $\max\{\text{dist}_{\mathcal{T}_1}(X_1, W), \text{dist}_{\mathcal{T}_1}(Y_1, W)\} \leq k$. De plus, du fait que Z est sur le chemin $\mathcal{T}_1[X_1, Y_1]$, on a soit $Z \in \mathcal{T}_1[W, X_1]$ soit $Z \in \mathcal{T}_1[W, Y_1]$. Supposons sans perdre de généralité que $Z \in \mathcal{T}_1[W, X_1]$. On a donc $\text{dist}_{\mathcal{T}_1}(W, Z) < \text{dist}_{\mathcal{T}_1}(X_1, W)$ car $Z \neq X_1$. Par conséquent, $\text{dist}_{\mathcal{T}}(W, Z) < \max\{\text{dist}_{\mathcal{T}}(X_1, W), \text{dist}_{\mathcal{T}}(Y_1, W)\} < k$. Ce qui veut dire que l'étape 2.1 n'a pas ajouté de nouvelles mauvaises paires de sacs distants de k ou plus. D'où, $|\mathcal{M}(\mathcal{T}_1)|$ diminue strictement. Or on sait que le nombre de paires de sacs possibles est $O(n^2)$ car $\mathcal{T}$ est réduite. Donc en temps $O(n^2)$ la procédure aura terminé de traiter les couples de sacs distants de k . De même pour les couples de sacs à distance strictement inférieure à k et ainsi de suite jusqu'à l'ensemble des couples de sacs à distance minimal > 0 . Par ailleurs, le nombre de classes de distances dans $\mathcal{T}$ entre ses sacs est au plus $O(n^2)$. On a montré donc que l'étape 2 termine en temps polynomial en n .

Enfin, montrons que l'étape 3 termine. En effet, il suffit de montrer que chaque sac de $\mathcal{T}_2$ est connexe car si c'est le cas, comme dans l'étape 1 on peut calculer les arbres couvrants pour chaque sac de $\mathcal{T}_2$ en temps polynomial.

Rappelons que dans la première partie, nous avons montré que s'il existait dans

$\mathcal{T}_2$ un sac X non connexe, alors cela contredirait la terminaison de l'étape 2. Donc, étant donné que l'étape 3 commence après que l'étape 2 soit terminée, alors chaque sac de $\mathcal{T}_2$ est connexe et donc l'étape 3 termine en temps polynomial. Nous avons donc montré que la procédure termine en temps polynomial. Ce qui termine la preuve du théorème 2.5.

2.4 Décompositions arborescentes des graphes planaires

Dans cette partie, nous présentons le résultat principal de ce chapitre. Cependant, avant de commencer, nous allons rappeler la propriété suivante ; elle est fondamentale pour notre algorithme.

Propriété 2.6 [*FG06, Lemme 12.10*]/[*Bod96, Tho04*] *Il existe un algorithme linéaire qui pour tout graphe planaire donne une décomposition arborescente dans laquelle tous les sacs sont composés de trois plus courts chemins.*

En particulier on a pour tout graphe planaire G de diamètre D , $\text{tw}(G) \leq 3D$.

Cependant, dans la preuve de [Bod96] les chemins sont de longueur $\Omega(D)$. L'une des conséquences de notre résultat est une généralisation de ce résultat puisque la longueur arborescente est toujours plus petite que le diamètre du graphe.

2.4.1 Définitions

Soit G un graphe de longueur arborescente au plus ℓ . Par le théorème 2.5, on peut supposer donnée $(\mathcal{T}, \mathcal{S})$, une décomposition arborée de G , 2-monotone et de profondeur $3\ell/2$. Pour tout couple de sacs $\{X, Y\}$ adjacents dans $\mathcal{T}$, notons par T_{XY} le sous-arbre minimal de T_X enraciné en $r(X)$ et couvrant $X \cap Y$.

Définition 2.7 (Diamant et sillon) *Pour tout couple de sacs (X, Y) adjacents dans $\mathcal{T}$, on définit par :*

- (i) *Le diamant de (X, Y) , noté $\diamond(X, Y)$, est le graphe induit par les sommets $r(X)$ et $r(Y)$, et tous les sommets w dont les chemins $T_X[w, r(X)]$ et $T_Y[w, r(Y)]$ utilisent au moins une arête de $T_{XY} \cup T_{YX}$.*
- (ii) *Le demi-diamant de X , noté $\triangle(X)$, est le graphe induit par $r(X)$ et tous les sommets qui ne sont dans aucun diamant impliquant X .*
- (iii) *Un sillon du diamant $\diamond(X, Y)$ est un chemin $P = T_X[r(X), z] \cup T_Y[z, r(Y)]$ avec $z \in X \cap Y$. Les chemins $p_1 = T_X[r(X), z]$ et $p_2 = T_Y[z, r(Y)]$ sont appelés des demi-sillons.*

Par exemple si on considère la décomposition arborescente $\mathcal{T}$ de figure 2.4 (a), on peut construire facilement les diamants formés au tour de chaque sac. En effet, le sac X matérialisé en bleu sur la figure est adjacent aux sacs Y, Z, W, R matérialisés respectivement en rouge, "gold", vert et noir. La figure 2.4 (b) met en évidence les diamants impliquant le sac X . On distingue les diamants $\diamond(X, Y)$, $\diamond(X, Z)$,

$\diamond(X, W)$ et $\diamond(X, R)$ formés à partir de l'intersection de l'arbre T_X avec respectivement les arbres T_Y , T_Z , T_W et T_R . Tous les diamants matérialisés ici sont disjoints à l'exception des diamants $\diamond(X, Y)$ et $\diamond(X, Z)$ qui partagent une branche de T_X . Nous avons choisi de ne matérialiser que les diamants impliquant X pour ne pas encombrer le schéma. Les diamants impliquant X sont organisés cycliquement autour de X comme on le verra dans la propriété 2.8 ci-dessous. Il peut arriver que des sommets appartenant à un sac X ne soient dans aucun des diamants impliquant X . Dans ce cas, la partie de l'arbre T_X couvrant ces sommets forme un demi-diamant. Sur la figure 2.4 (b), le sac X n'a pas de demi-diamant alors que le sac Z en a un qui est découpé en deux par les diamants impliquant Z .

Nous avons aussi matérialisé sur le diamant $\diamond(X, Z)$ de la figure 2.4 (b), deux sillons consécutifs de diamant dont l'intérieur est coloré en bleu.

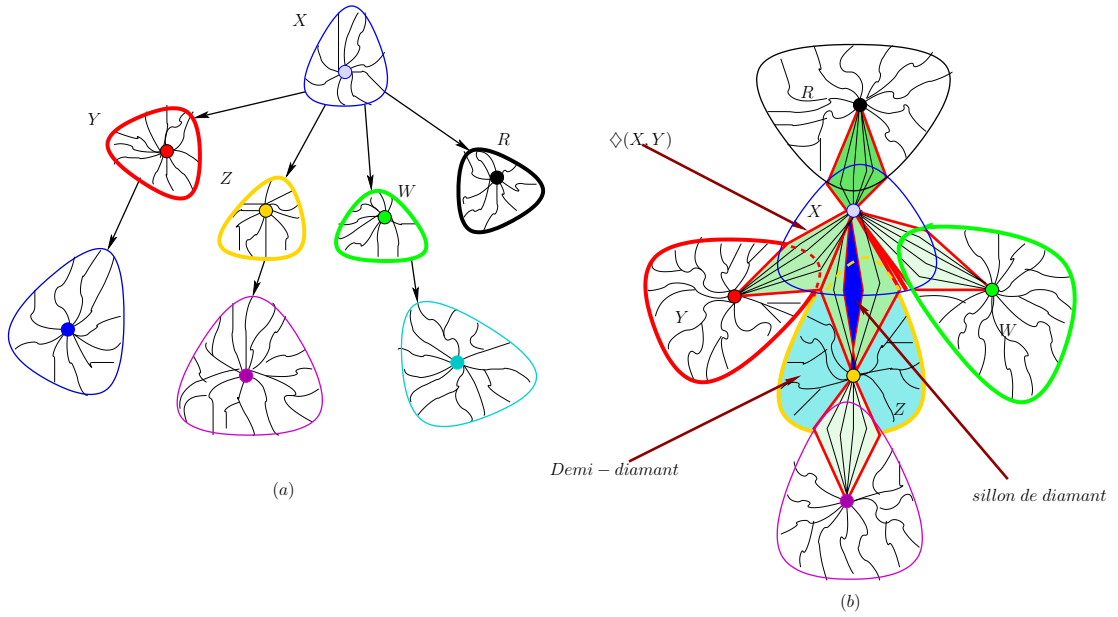


FIG. 2.4: (a) Une décomposition arborescente enracinée $\mathcal{T}$. (b) Un sac X et 4 sacs Y, Z, W, R adjacents à X dans $\mathcal{T}$ et une représentation des diamants impliquant X .

On a la propriété suivante :

Propriété 2.8 Soit G un graphe planaire, $(\mathcal{T}, \mathcal{S})$ une décomposition arborée de G , et Y et Z deux voisins distincts de X dans $\mathcal{T}$. Alors T_{XY} et T_{XZ} s'intersectent en au plus deux chemins partant de $r(X)$.

Démonstration : Soient X, Y et $Z \in \mathcal{T}$ tels que Y et Z sont voisins de X dans $\mathcal{T}$. Supposons que T_{XY} intersecte T_{XZ} en plus de deux chemins partant de $r(X)$.

Posons v_1, v_2 et v_3 trois sommets de $T_{XY} \cap T_{XZ}$ qui sont sur trois chemins différents partant de $r(X)$. Ces trois sommets existent car le fait que T_X et T_Y soient enracinés dans des parties privées implique que $r(X)$ et $r(Y)$ ne sont pas adjacents, et donc il existe dans $T_{XY} \cup T_{YX}$ un chemin de longueur au moins deux.

Donc, le sous graphe induit par les chemins $T_{XY}[r(X), v_1]$, $T_{XY}[r(X), v_2]$, $T_{XY}[r(X), v_3]$, $T_{YX}[r(Y), v_1]$, $T_{YX}[r(Y), v_2]$, $T_{YX}[r(Y), v_3]$, $T_{ZX}[r(Z), v_1]$, $T_{ZX}[r(Z), v_2]$, $T_{ZX}[r(Z), v_3]$ contient $K_{3,3}$ comme mineur. Ce qui est une contradiction avec le fait que G est sans mineur $K_{3,3}$. Ce qui termine la preuve de la propriété 2.8. $\square$

Cela fait qu'on a deux types de diamants. Il y a les diamants de type un qui intersectent tout autre diamant en au plus un chemin et les diamants de type deux s'intersectant en deux chemins. Les diamants qui n'intersectent aucun autre diamant sont de type deux car il intersectent tout autre diamant en zéro chemin c'est dire au plus un. Par exemple, sur la figure 2.4 tous les diamants matérialisés sont de type un. Le diamant impliquant X et R n'intersecte aucun autre diamant ; il est de type un. De même, les autres diamants impliquant X intersectent tout autre diamant en un chemin ; ils sont donc de type un.

2.4.2 Décomposition arborescente de graphes planaires

Dans cette partie nous allons discuter du principal résultat de ce chapitre. Nous allons en fait montrer le résultat suivant.

Théorème 2.9 *Tout graphe planaire valué ayant une décomposition arborescente de longueur ℓ admet une décomposition arborescente dans laquelle chaque sac est l'union d'au plus 12 plus courts chemins de longueur au plus ℓ .*

Démonstration : Soit G un graphe planaire de longueur arborescente ℓ . Par le théorème 2.5, on peut supposer $(\mathcal{T}, \mathcal{S})$, une décomposition arborée de G , 2-monotone et de profondeur $3\ell/2$ et tel que $\mathcal{T}$ est enraciné en un sac $R \in \mathcal{T}$.

Grâce à la procédure **MAIN1**($\mathcal{T}, R, \mathcal{T}_1$), on va construire à partir de $\mathcal{T}$ un arbre de décomposition $\mathcal{T}_1$ de G dans lequel chaque sac X de $\mathcal{T}_1$ contient soit un diamant, soit une composante connexe issue d'un demi-diamant soit un ensemble de sommets x tel qu'il existe deux sacs Y et Z adjacents à X et $x \in Y \cap Z$.

Ensuite, grâce à la procédure **MAIN**($\mathcal{T}, X$), on transforme l'arbre $\mathcal{T}_1$ en un arbre de décomposition de G dans lequel chaque sac X possède au plus 6 chemins de longueur au plus $3\ell/2$, composé chacun d'au plus deux plus courts chemins.

Initialement on lance **MAIN1**($\mathcal{T}, R, \mathcal{T}_1$), ou $\mathcal{T}_1$ est égal à $\emptyset$. Ensuite on lance **MAIN**($\mathcal{T}_1, R$) avec $\mathcal{T}_1$ l'arbre produit par la procédure **MAIN1**. Dans toute la suite pour tout sac Y de $\mathcal{T}_1$, on notera par $\mathcal{T}_{1Y}$ le sous-arbre de $\mathcal{T}_1$ enraciné en Y .

MAIN1($\mathcal{T}, X, \mathcal{T}_1$) :

1. Poser $\mathcal{S}_X$ un ensemble de sacs tels que chaque sac de $\mathcal{S}_X$ contient un diamant impliquant X . Soit $P \in \mathcal{T}_1$ le sac contenant le diamant impliquant X et $père(X)$. Posons R_0 un sac contenant éventuellement, le demi-diamant de X , ainsi que l'ensemble des demi-sillons couvrant des sommets de X intersectant deux diamants impliquant X , plus l'ensemble des couples de demi-sillons (s_1, s_2) appartenant à deux diamants différents de $\mathcal{S}_X$ et tel que $\exists x \in s_1$ et $y \in s_2$, et $(x, y) \in E(G)$. Pour chaque sac $X_0 \in \mathcal{S}_X$, faire si $X_0 \neq P$, alors $père(X_0) = R_0$. Faire $père(R_0) = P$.
2. Pour tout couple $\{Y, Z\} \in \mathcal{S}_X$, tel que $Y \cap Z$ est sur deux sillons de R_0 et que $Y, Z \neq P$, faire : arbitrairement $père(Z) = Y$.
 - Tant que $(\exists Z_0 \in \mathcal{T}_{1Z}, \exists W \in S$ tel que $W \cap Z_0 \neq \emptyset$ et $père(W) = R_0$) faire : $père(W) = Z_0$.
 Pour tout sac $Z_0 \in \mathcal{T}_{1Z}$ tel que $P \cap Z_0 \neq \emptyset$ faire : pour tout sac $R_i \in \mathcal{T}_1[P, Z_0]$, $R_i = R_i \cup \{P \cap Z\}$.
3. Si R_0 contient un demi-diamant $\Delta(X)$, alors poser S' un ensemble de sacs tels que chaque sac contient une composante connexe du sous-graphe induit par $G[\Delta(X)] \setminus r(X)$ plus $r(X)$.
 - 3.1. Pour tout sac $Y \in S'$, s'il existe un sac $Z \in S$ tel que $Y \cap S \subseteq Z$, alors faire $père(Y) = Z$, $R_0 = R_0 \setminus Y$.
 - 3.2. Pour tout sac $Y \in S'$ tel que $père(Y) = \emptyset$ faire $père(Y) = R_0$, $R_0 = R_0 \setminus Y$ plus l'ensemble des sillons de X couvrant $Y \cap \mathcal{S}_X$.
 Pour chaque sac $Y \neq père(X)$ adjacent à X dans $\mathcal{T}$, faire **MAIN1**($\mathcal{T}_Y, Y, \mathcal{T}_1$).

Lemme 2.10 *La procédure **MAIN1** produit un arbre $\mathcal{T}_1$ de décomposition arborescente de G dans laquelle chaque sac X de $\mathcal{T}_1$ contient soit un diamant, soit une composante connexe issue d'un demi-diamant, soit un ensemble de sillons couvrant l'intersection entre deux sacs adjacents à X .*

Démonstration : Par construction de l'arbre $\mathcal{T}_1$, chaque sac X de $\mathcal{T}_1$ contient soit un diamant, soit une composante connexe issue d'un demi-diamant, soit un ensemble de sillons couvrant l'intersection entre deux sacs adjacents à X .

Pour montrer que l'arbre $\mathcal{T}_1$ produit par la procédure **MAIN1** est un arbre de décomposition arborescente de G , nous allons montrer qu'après chacune des étapes de la procédure **MAIN1**, l'arbre produit vérifie les trois conditions de la définition 2.1. Dans toute la suite de la preuve, $\mathcal{T}_i$ (pour $1 \leq i \leq 3$) représente l'arbre $\mathcal{T}_1$ après l'étape i .

Vérification des trois conditions de la définition après l'étape 1 : La première étape de la procédure **MAIN1** consiste dans un premier temps à prendre le sac $X \in \mathcal{T}$ à traiter, d'identifier l'ensemble $\mathcal{S}_X$ des diamants impliquant X et à mettre chaque diamant de $\mathcal{S}_X$ dans un sac. Dans un deuxième temps, on identifie l'ensemble des sommets de X qui n'appartiennent à aucun sac de $\mathcal{S}_X$ et on les met

dans un sac nommé R_0 ainsi que l'ensemble des demi-sillons couvrant l'intersection entre tout couple de diamants $\{Y, Z\}$ de $\mathcal{S}_X$. De plus, on met dans R_0 l'ensemble des couples de demi-sillons (s_1, s_2) de X , appartenant respectivement à Z_{s_1} et Z_{s_2} deux diamants de $\mathcal{S}_X$ tels que $Z_{s_1} \neq Z_{s_2}$ et il existe deux sommets $x \in s_1$ et $y \in s_2$ tels que l'arête $(x, y) \in E(G)$. Enfin, on intègre les nouveaux sacs ainsi obtenus dans $\mathcal{T}_1$. En effet, à l'exception du sac P contenant le diamant impliquant X et $\text{père}(X)$ dans $\mathcal{T}$, tous les sacs de $\mathcal{S}_X$ ont comme père le sac R_0 . Vu que la procédure est supposée être initialement appelée à la racine de $\mathcal{T}$, alors il est évident que le diamant impliquant X et $\text{père}(X)$ est déjà dans un sac de $\mathcal{T}_1$. En d'autres termes, le sac P est déjà connecté à $\mathcal{T}_1$. Ainsi, en supposant que R_0 a comme père le sac P , on a rattaché tous les sacs de $\mathcal{S}_X$ à $\mathcal{T}_1$. Dans le cas où X est la racine de $\mathcal{T}$, cela reste toujours vrai car ce cas équivaut à $P = \emptyset$ et dans ce cas R_0 est la racine de $\mathcal{T}_1$. Cependant, il faut remarquer que le sac R_0 contient non seulement le demi-diamant impliquant X mais aussi les intersections de diamants impliquant X . La figure 2.5 (b) est un exemple d'exécution de l'étape 1 sur l'arbre de décomposition de la figure 2.5 (a). Dans cet exemple, on remarque que le sac X représenté en bleu sur la figure 2.5 (a) est la racine de $\mathcal{T}$ et l'ensemble des sacs de $\mathcal{T}_1$ couvrant $\mathcal{S}_X$ a la structure d'une étoile matérialisée par une patate bleue dans l'arbre $\mathcal{T}_1$ de la figure 2.5 (b). Ainsi, le centre de cette étoile contient non seulement les demi-sillons (matérialisé par des segments rouges sur la figure 2.5 (b)) intersectant les diamants de l'étoile, mais aussi un ensemble de sommets qui ne sont dans aucun diamant de l'étoile. Ces sommets sont matérialisés par des segments noirs; ce sont des branches de l'arbre $T_{r(X)}$. On peut aussi remarquer que tous les sacs de cette étoile contiennent le sommet $r(X)$ matérialisé en bleu. Cet exemple illustre bien le cas où le sac X à traiter est la racine de $\mathcal{T}$. Ce même raisonnement peut être appliqué pour illustrer les autres sacs en descendant dans $\mathcal{T}_1$. Sur la figure 2.5 (b), chaque étoile de $\mathcal{T}_1$ est représentée par une patate ayant la même couleur que le sac représentant cette étoile dans $\mathcal{T}$. Il est clair qu'après cette première étape, l'arbre $\mathcal{T}_1$ obtenu vérifie bien les conditions de la définition 2.1. En effet, tout sommet $x \in X$ est dans au moins un sac de $\mathcal{S}_X$. De plus, pour tout couple de sommets $(x, y) \in X$ tel que x adjacent à y dans G , on a soit x et y sont tous les deux dans un diamant ou demi-diamant, soit ils sont chacun dans un diamant ou demi-diamant. Dans le premier cas, il existe un sac de $\mathcal{S}_X$ contenant le diamant de x et y et donc la condition (ii) est vérifiée. Si les deux sommets sont dans deux diamants différents, alors ils sont dans R_0 et donc l'arbre $\mathcal{T}_1$ obtenu vérifie la condition (ii) de la définition 2.1. Nous pouvons aussi remarquer sans ambiguïté que l'étape 1 peut se faire en temps polynomial en $|\mathcal{S}_X|$.

Vérification des trois conditions de la définition après l'étape 2 : Il faut remarquer que l'arbre que nous voulons produire est tel que chaque composante connexe issue $\Delta(X) \setminus r(X)$ est dans un sac. C'est la tâche à réaliser par l'étape 3. L'étape 2 est nécessaire pour des ajustements techniques dans le cas où on manipule des diamants de type 2. Sur l'exemple de la figure 2.5, on remarque l'existence de deux diamants de type 2 : celui impliquant W_1 et W_2 , et celui entre W_1 et W_3 . Nous remarquons que dans la décomposition avec l'étape 1, il n'y a pas de différence de traitement selon le type.

Le but de l'étape 3 est d'une part de procéder à un redécoupage du demi-diamant couvert par R_0 mais aussi de faire des corrections dans R_0 . C'est aussi dans l'étape 3 qu'on fait un appel récursif sur chaque sous-arbre $\mathcal{T}_Z$ de $\mathcal{T}$ enraciné en un sac Z adjacent à X dans $\mathcal{T}$.

Rappelons que l'étape 2 consiste à détecter dans $\mathcal{S}_X$ les sacs contenant un diamant de type 2. En effet, pour tout couple $\{Y, Z\}$ de sacs de $\mathcal{S}_X$ tel que $Y \cap Z$ est sur deux demi-sillons de X , alors Y et Z contiennent deux diamants de type 2. Dans ce cas, on choisit arbitrairement entre Y et Z le sac qui va devenir le père de l'autre. On peut supposer sans perdre de généralité que Y est choisi comme étant le père de (Z) . Une fois que le père de Z est modifié, on a le sous-arbre $\mathcal{T}_{1Z}$ qui est constitué d'un seul sommet : le sommet Z . On procède ainsi à la recherche de l'existence d'un sac $W \in N(R_0)$ tel que $W \cap Z \neq \emptyset$. Si W existe, on entre dans la boucle "tant que" et on modifie le père de W qui va devenir Z . Une fois cela fait, le sous-arbre $\mathcal{T}_{1Z}$ est constitué de deux sommets : Z et W . Ainsi, on recherche à nouveau si il existe un sac $W' \in N(R_0)$ tel que $W' \cap Z \neq \emptyset$ ou bien $W' \cap W \neq \emptyset$. Si W' existe, on change le père de W' en Z ou bien W . On réitère ce procédé autant de fois que possible. En d'autres termes, on le réitère au plus $|\mathcal{S}_X|$ fois. Ce qui prouve donc la terminaison de l'étape 2 en temps polynomial en n car $|\mathcal{S}_X| \leq n$.

Il est clair que si $\mathcal{T}_1$ vérifie les conditions (i) et (ii) de la définition 2.1 avant l'étape 2, alors l'arbre $\mathcal{T}_2$ produit après l'étape 2 le vérifie. En effet, les seules opérations effectuées dans l'étape 2 sont des déplacements de sacs. Donc, si toute arête et tout sommet de G se trouve dans un sac avant l'étape 2, il reste dans ce même sac après l'étape 2.

Supposons que l'arbre $\mathcal{T}_2$ produit après l'étape 2 ne vérifie pas la condition (iii) de la définition 2.1. Vu qu'après l'étape 1, l'arbre $\mathcal{T}_1$ vérifiait toutes les conditions de cette définition, alors il existe au moins un sac adjacent à R_0 dont on a modifié le père et qu'après cette modification, la condition n'est plus vérifiée. Soit W un sac dont le père est modifié dans l'étape 2 et soit Z le nouveau père. Il existe donc deux sacs W' et R tels que $R \in \mathcal{T}_1[W, W']$, $W \cap W' \not\subseteq R$ et $W \cap W' \neq \emptyset$.

Alors vu que l'arbre $\mathcal{T}_1$ vérifiait les conditions de la définition 2.1 avant l'étape 2, on a $R \in \mathcal{T}[Z, R_0]$. En effet, si on avait $R \in \mathcal{T}_1[R_0, W']$, alors on aurait $R \in \mathcal{T}_1[W', W]$ avant même l'application de l'étape 2 et donc la condition (iii) ne serait pas respectée par $\mathcal{T}_1$; ce qui est une contradiction. Par conséquent, on a pour tout sac $R_i \in \mathcal{T}_1[W', R_0]$, $W \cap W' \subseteq R_i$. Donc, $\forall R_i \in \mathcal{T}_1[W', R_0]$, il existe $R \in \mathcal{T}_1[W, R_i]$ tel que $W \cap R_i \not\subseteq R$ et $W \cap R_i \neq \emptyset$. Nous avons donc $W \in \mathcal{T}_Z$ et $W' \in N(R_0)$ tel que $W \cap W' \neq \emptyset$; ce qui est une contradiction avec la terminaison de l'étape 2. Donc, l'arbre $\mathcal{T}_2$ vérifie bien la condition (iii) de la définition 2.1.

Dans l'exemple de la figure 2.6, l'étape 2 va se dérouler uniquement pour la décomposition du sac W_1 de $\mathcal{T}$. En effet, ce sac est impliqué dans deux diamants de type 2 : le diamant formé entre W_1 et W_2 et celui formé entre W_1 et W_3 . On remarque dans $\mathcal{T}_1$ qu'après l'application de l'étape 1 sur le sac W_1 , on a le sac R_{0W_1} qui est adjacent à deux sacs qui s'intersectent en deux demi-sillons de W_1 . L'étape 2 change donc le père de l'un des deux sacs; ici c'est le sac contenant le diamant $\diamond(W_1, W_3)$ bleu qui devient le père du sac contenant le diamant $\diamond(W_1, W_2)$.

Vérification des trois conditions de la définition après l'étape 3 : L'étape 3 consiste à redécomposer le demi-diamant contenu dans le sac R_0 . En effet, pour tout demi-diamant $\Delta(X)$ de pôle $r(X)$, on pose S un ensemble de sacs dont chacun contient une composante connexe engendrée par $G[\Delta(X) \setminus r(X)]$ plus $r(X)$. Ensuite, pour tout sac $Y \in S$, on regarde s'il existe un sac $Z \in \mathcal{S}_X$ tel que $Y \cap \mathcal{S}_X \setminus \{r(X)\} \subseteq Z$. Si Z existe, alors on décide que Z devient le père de Y . Sinon, on décide que R_0 devienne le père de Y . Il est clair que la seule condition qui peut ne pas être vérifiée après cette action est la condition (iii) de la définition 2.1 car nous ne faisons pas d'action sur les sommets mais sur les sacs.

Supposons que la condition (iii) de définition 2.1 ne soit pas vérifiée par $\mathcal{T}$ après l'étape 3. Soit Y le premier sac créé dans l'étape 3 et qui après l'avoir connecté à $\mathcal{T}_1$, la condition (iii) ne soit plus vérifiée par $\mathcal{T}_1$. Il existe donc deux sacs Z et $R \in \mathcal{T}_1$ tels que $Z \in \mathcal{T}_1[Y, R]$ et $Y \cap R \not\subseteq Z$. Donc Y n'est pas adjacent à R_0 car sinon, les sacs R_0 et R ne vérifieraient pas la condition (iii) avant même la création de Y ; ce qui est une contradiction avec l'hypothèse. Or, par définition de l'étape 3, Y est non adjacent à R_0 si et seulement si $\exists Z_0 \in \mathcal{T}_1$ tel que $Y \cap \mathcal{S}_X \subseteq Z_0$. Donc, on a $R \cap Y \subseteq Z_0$. D'où, on a $Z_0 \cap R \not\subseteq Z$; ce qui contredit le fait que Y est le premier sac créé dans l'étape 3 et qui après l'avoir connecté à $\mathcal{T}_1$, la condition (iii) ne soit plus vérifiée par $\mathcal{T}_1$. Par conséquent, $\mathcal{T}_3$ vérifie toutes les conditions de la définition 2.1. Ce qui termine la preuve du lemme 2.10. $\square$

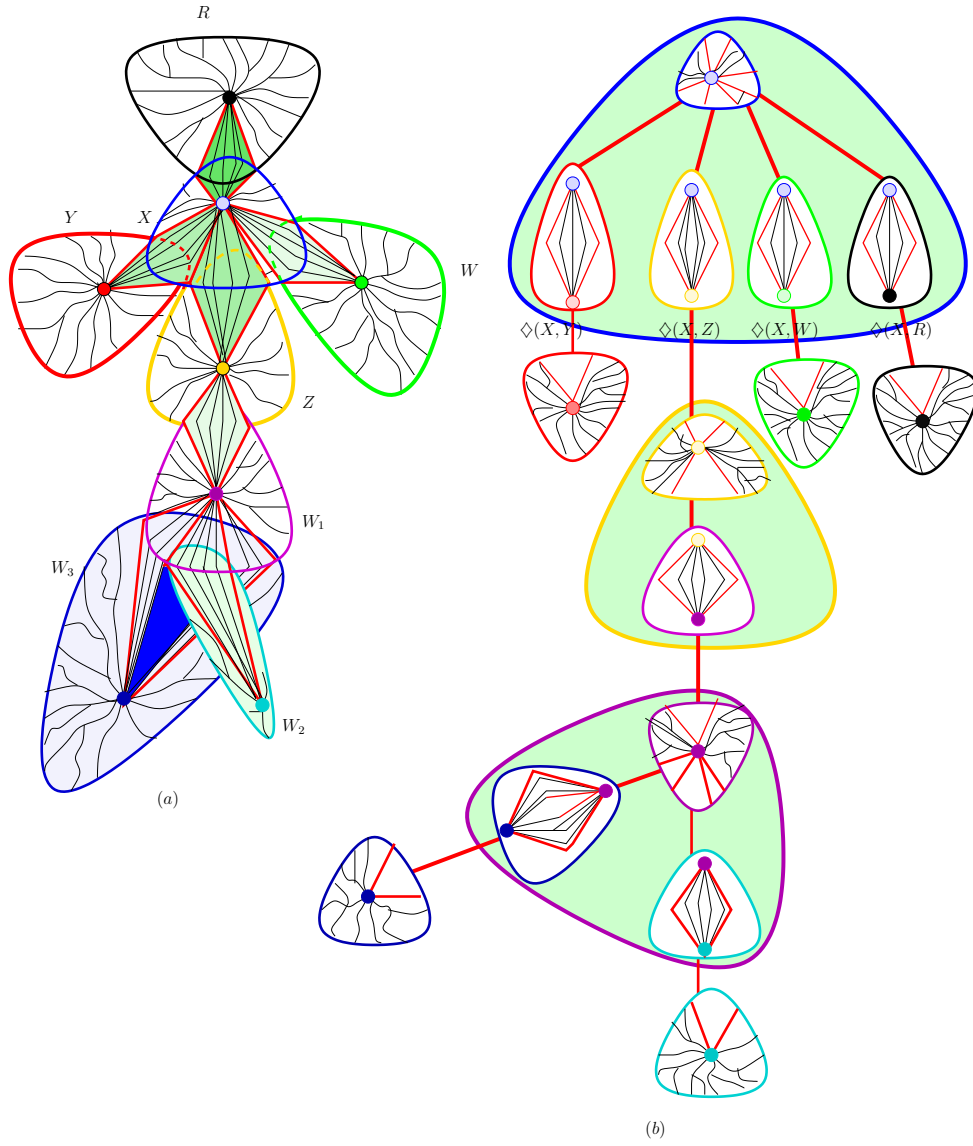


FIG. 2.5: (a) Un arbre $\mathcal{T}$ d'une décomposition arborée $(\mathcal{T}, \mathcal{S})$, enraciné en un sac X adjacent à 4 sacs Y, Z, W, R . (b) Une décomposition arborescente $\mathcal{T}_1$ obtenue après l'application de l'étape 1 de la procédure **MAIN1** $(\mathcal{T}_X, X, \mathcal{T}_1 = \emptyset)$ à chaque sac X de $\mathcal{T}$.

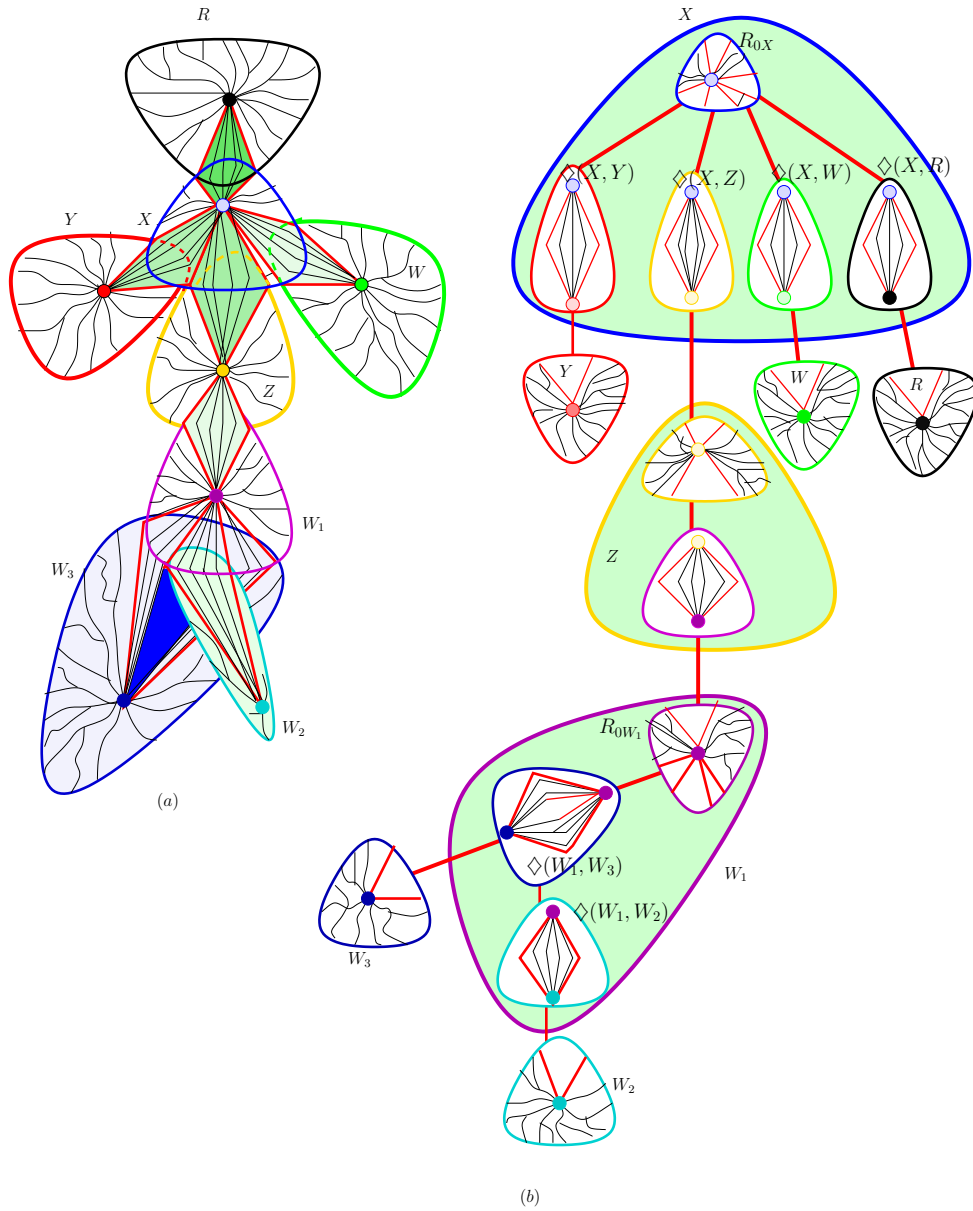


FIG. 2.6: (a) Un arbre $\mathcal{T}$ d'une décomposition arborée $(\mathcal{T}, \mathcal{S})$, enraciné en un sac X adjacent à 4 sacs Y, Z, W, R . (b) Une décomposition arborescente $\mathcal{T}_1$ obtenue après l'application des étape 1 et 2 de la procédure **MAIN1**($\mathcal{T}_X, X, \mathcal{T}_1 = \emptyset$) à chaque sac X de $\mathcal{T}$.

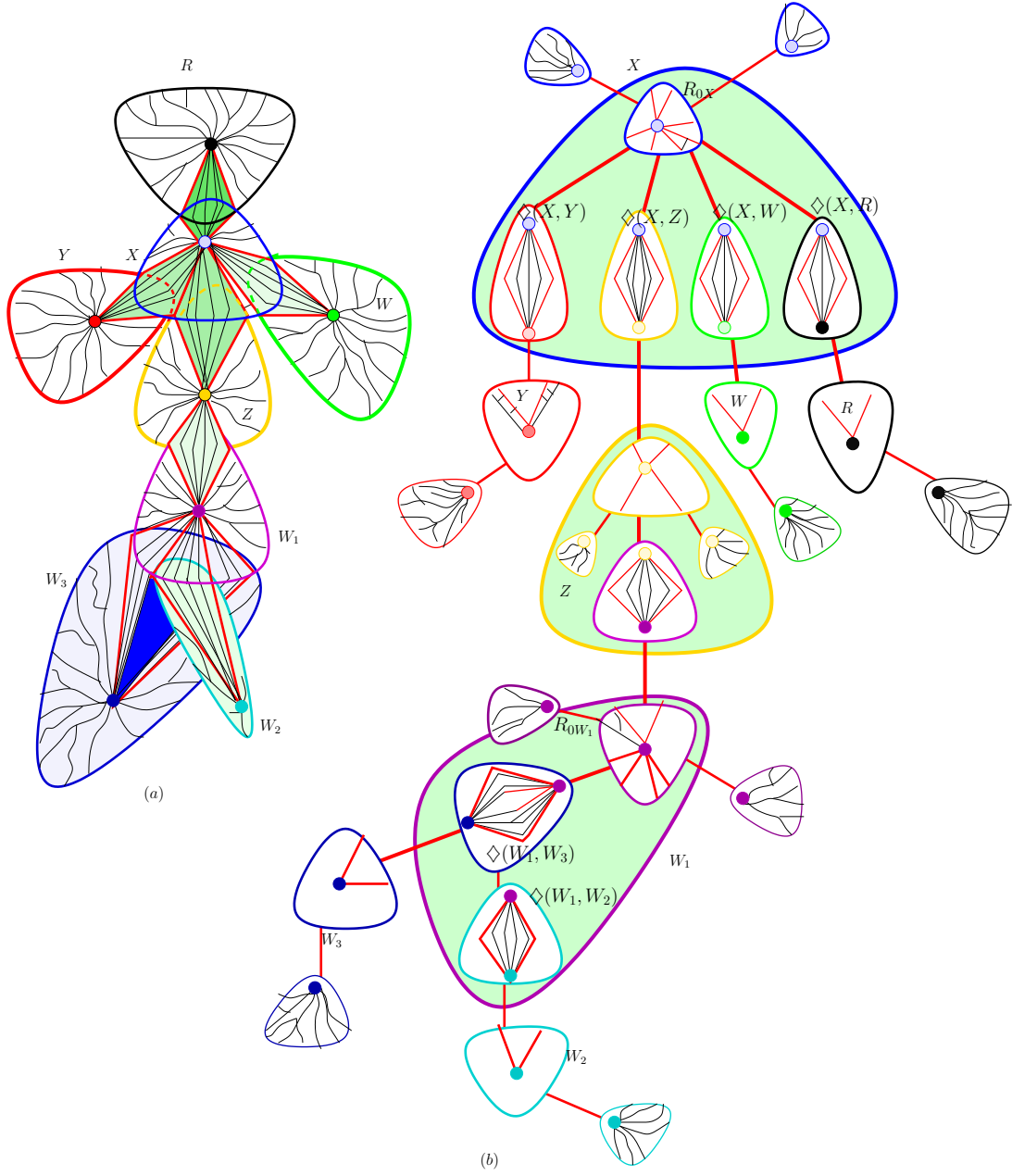


FIG. 2.7: (a) Un arbre $\mathcal{T}$ d'une décomposition arborée $(\mathcal{T}, \mathcal{S})$, enraciné en un sac X adjacent à 4 sacs Y, Z, W, R . (b) Une décomposition arborescente $\mathcal{T}_1$ obtenue après l'application de la procédure **MAIN1**($\mathcal{T}_X, X, \mathcal{T}_1 = \emptyset$).

Nous allons maintenant, grâce à la procédure **MAIN**, modifier l'arbre $\mathcal{T}_1$ de telle sorte que chaque sac possède au plus 6 chemins de longueur au plus $3\ell/2$.

Pour tout sac X de $\mathcal{T}_1$ contenant un diamant de type 1, soit **DecomposerType1**($\mathcal{T}_1, X$) la procédure qui remplace X dans $\mathcal{T}_1$ par la racine R d'un arbre de décomposition arborescente $\mathcal{T}_0$ construit en redécomposant le sac X tel que chaque nœud de $\mathcal{T}_0$ contient au plus 6 chemins de X de longueur $\leq 3\ell/2$.

Pour tout sac X de $\mathcal{T}_1$ contenant un diamant de type 2, soit **DecomposerType2**($\mathcal{T}_1, X$) une procédure qui remplace X dans $\mathcal{T}_1$ par la racine R d'un arbre $\mathcal{T}_0$ construit en redécomposant le sac X tel que chaque nœud de $\mathcal{T}_0$ contient au plus 6 chemins de X de longueur $\leq 3\ell/2$.

Pour tout sac X de $\mathcal{T}_1$ contenant une composante connexe issue d'un demi-diamant, soit **DECOMPOSER**($\mathcal{T}_1, X$) une procédure qui remplace X dans $\mathcal{T}_1$ par la racine R d'un arbre $\mathcal{T}_0$ construit en redécomposant le sac X tel que chaque nœud de $\mathcal{T}_0$ contient au plus 3 chemins de X de longueur $\leq 3\ell/2$.

Pour tout sac X de $\mathcal{T}_1$ contenant des sillons intersectant X avec $N(X)$, soit **DecomposerIntersection**($\mathcal{T}_1, X$) une procédure qui remplace le sac X dans $\mathcal{T}_1$ par la racine R d'un arbre $\mathcal{T}_0$ construit en redécomposant le sac X tel que chaque nœud de $\mathcal{T}_0$ contient au plus 5 chemins de X de longueur $\leq 3\ell/2$.

MAIN($\mathcal{T}, R$) :

1. Pour tout fils Z de R , faire : **MAIN**($\mathcal{T}_Z, Z$)
2. Si R contient un diamant de type 1, alors faire : **DecomposerType1**($\mathcal{T}_R, R$)
3. Sinon, si R contient un diamant de type 2, alors **DecomposerType2**($\mathcal{T}_R, R$)
4. Sinon, si R contient une composante connexe C d'un demi-diamant $\Delta(X)$, alors posons $r(X)$ le pôle de $\Delta(X)$. Soit H le graphe obtenu à partir de $G[C]$ en remplaçant chaque couple (s_1, s_2) de sillons de R intersectant un même sac de $N(R)$ par $\max\{s_1, s_2\}$ et marquer ce sillon. Faire $(W, \mathcal{T}_0) = \mathbf{DECOMPOSER}(H)$. Pour chaque sac X de $\mathcal{T}_0$ contenant un sillon s marqué, dupliquer s dans X .
Pour chaque sac W' adjacent à R dans $\mathcal{T}$, chercher dans $\mathcal{T}_0$ le premier sac $Z_0 \in \mathcal{T}_0$ partant de la racine de $\mathcal{T}_0$ tel que $W' \cap Z_0 \neq \emptyset$ et faire : si $\text{père}(W') = R$, alors $\text{père}(W') = Z_0$ sinon $\text{père}(Z_0) = W'$.
5. Sinon, **DecomposerIntersection**($\mathcal{T}_R, R$).

Il est clair que si $\mathcal{T}$ est produit par la procédure **MAIN1**, alors tout sac X de $\mathcal{T}$ contient ou un diamant qui est de type 1 ou 2, ou bien contient un ensemble de sommets d'une composante connexe issue d'un demi-diamant, ou contient un ensemble de sillons contenant l'intersection entre X et tout sac Y adjacent à X dans $\mathcal{T}$. Or, on sait que dans chacun des cas, on a un algorithme qui remplace X par la racine R d'un arbre $\mathcal{T}_0$ dans lequel chaque sac contient au plus 6 chemins de longueur $\leq 3\ell/2$. Il est facile de voir que si R contient $X \cap \text{père}(X)$, alors l'arbre produit est un arbre de décomposition car $\mathcal{T}_0$ est par hypothèse un arbre de décomposition. Par consé-

quent, en appliquant la procédure **MAIN**($\mathcal{T}_1, R$), on obtient une décomposition $\mathcal{T}_1$ de G dans laquelle chaque sac possède au plus 6 chemins de longueur $\leq 3\ell/2$.

DecomposerType1($\mathcal{T}_R, R$) :

1. Soit R_1 un sac contenant l'ensemble des sillons de R couvrant $R \cap N(R)$ plus un sillon s de R tel que $s \not\subseteq R_1$ s'il existe, sinon $s = \emptyset$.
2. Faire $\text{père}(R_1) = \text{père}(R)$ et pour chaque fils X de R dans $\mathcal{T}_R$, faire $\text{père}(X) = (R_1)$. Mettre chaque composante connexe de $G[R \setminus R_1]$ dans un sac R' et faire :
 - a) $\text{père}(R') = R_1$.
 - b) **DecomposerType1**($\mathcal{T}_{R'}, R'$).
3. Supprimer R et faire **DecomposerIntersection**($\mathcal{T}_{R_1}, R_1$).

Montrons que **DecomposerType1**($\mathcal{T}, R$) remplace R par un arbre $\mathcal{T}_0$ de décomposition de R . En effet, la première étape construit un sac R_1 dans lequel il met tous les sillons de R qui contiennent au moins un sommet x appartenant à la fois à R et à au moins un sac Z adjacent à R . Ensuite, on ajoute à R_1 un sillon qui contient uniquement des sommets propres à R .

On remplace ainsi le sac R par le sac R_1 et on ré-applique la procédure sur tous les fils R' de R_1 engendré chacun par une composante connexe de $G[R \setminus R_1]$. Enfin, pour redécouper le sac R_1 dans le cas où il contient plus de 3 sillons, on appelle la procédure **DecomposerIntersection**($\mathcal{T}_{R_1}, R_1$). Ainsi, les sacs $\mathcal{T}$ issus du sac R contiennent au plus 3 sillons et donc 6 chemins car chaque sillon est par définition constitué par deux chemins de longueur $\leq 3\ell/2$.

De plus, il est clair que la terminaison de la procédure dépend de celle de **DecomposerIntersection**($\mathcal{T}_{R_1}, R_1$). En effet, les instructions de la procédure **DecomposerType1** peuvent être faites en temps polynomial en n . C'est juste la recherche d'un sommet intersectant R et un sac adjacent à R .

DecomposerType2($\mathcal{T}_R, R$) :

1. Soit $P =$ l'ensemble des sillons de R couvrant $R \cap N(R)$. Soit $\mathcal{C} =$ l'ensemble des composantes connexes de $G[R \setminus P]$.
2. Mettre chaque composante connexe C de $\mathcal{C}$ dans un sac R_C et faire $R = P$, $\text{père}(R_C) = (R)$ et **DecomposerType1**($\mathcal{T}_{R_C}, R_C$)
3. **DecomposerIntersection**($\mathcal{T}_R, R$)

Montrons que **DecomposerType2**($\mathcal{T}, R$) remplace R par un arbre $\mathcal{T}_0$ de décomposition de R . En effet, comme la procédure **DecomposerType1**($\mathcal{T}, R$), remplace R par un sac W ne contenant que l'intersection entre R et $N(R)$. Ensuite chaque composante connexe de $G[R \setminus P]$ (P est l'ensemble de sillons de R couvrant $R \cap N(R)$) est remplacé par un arbre $\mathcal{T}_0$ dont chaque sac contient au plus 3 sillons grâce à la procédure **DecomposerType1**. Enfin, le sac W qui remplace R est de nouveau remplacé par un arbre $\mathcal{T}_1$ dans lequel chaque sac contient au plus 5 chemins grâce à

la procédure **DecomposerIntersection** qui est invoquée sur W . Nous avons donc remplacé le sac R par un arbre de décomposition dans lequel chaque sac possède au plus 6 chemins de longueur $\leq 3\ell/2$.

De plus la terminaison de cette procédure dépend uniquement de celles des procédures **DecomposerType1**($\mathcal{T}, R$) et **DecomposerIntersection**. En effet, les instructions dans cette procédure consistent à calculer les sillons qui intersectent R et à déterminer les composantes connexes de $G[R \setminus P]$. Ce sont des opérations qui se font en temps polynomial.

À partir de maintenant, pour tout diamant $\diamond(X, Y)$ engendré par deux sacs X et Y d'une décomposition arborée $(\mathcal{T}, \mathcal{S})$, on appellera *pôles* du diamant les sommets $r(X)$ et $r(Y)$. Notons que si c'est un demi-diamant, alors on a un seul pôle.

DecomposerIntersection($\mathcal{T}_R, R$) :

1. Si R contient un diamant, alors posons $r(X)$ et $r(Y)$ les deux pôles du diamant ; sinon R contient un demi-diamant et soit $r(X)$ le pôle de R . Pour chaque sac Z , adjacent à R , mettre les sillons (ou chemins) couvrant $R \cap Z$ dans un sac noté R_Z et soit $\mathcal{S}$ l'ensemble des sacs ainsi obtenus. Posons $R_0 = R \setminus \mathcal{S}$. Pour chaque sac de $\mathcal{S}$, s'il existe un sillon (ou chemin) s tel que $N(s) \cap R_0 \neq \emptyset$, alors mettre s et les sillons couvrant $N(s) \cap R_0$ dans un sac R_s et ajouter R_s à $\mathcal{S}$. Pour chaque sac $R_z \in \mathcal{S}$ engendré par $Z \in N(R)$, si $Z \neq \text{père}(R)$ faire $\text{père}(Z) = R_Z$. Posons $P = \text{père}(R)$, et R_P le sac de $\mathcal{S}$ contenant $R \cap P$. Remplacer R par R_P en faisant $\text{père}(R_P) = P$, marquer R_P comme une feuille de $\mathcal{T}_R$.
2. Tant que : $(\exists Y \in \mathcal{S} \text{ tel que } \text{père}(Y) = \emptyset)$ alors faire :
 - 2.1 Pour toute feuille Y de $\mathcal{T}_R$ faire :
 - 2.1.1 Pour tout sac $Z \in \mathcal{S}$ tel que $\text{père}(Z) = \emptyset$ et $Y \cap Z \setminus \{r(X), r(Y)\} \neq \emptyset$ faire : $\text{père}(Z) = Y$ et marquer Z comme feuille.
 - 2.1.2 Marquer Y comme un sac simple.
 - 2.2 S'il existe $Z \in \mathcal{S}$ tel que $\text{père}(Z) = \emptyset$ et que pour tout $Z \in \mathcal{S}$ tel que $\text{père}(Z) = \emptyset$ et pour toute feuille $Y \in \mathcal{T}_R$, on a $Y \cap Z \setminus \{r(X), r(Y)\} = \emptyset$, alors choisir arbitrairement $Z \in \mathcal{S}$ et faire $\text{père}(Z) = R_P$, marquer Z comme feuille.
3. Pour tout couple $\{Z_f, Z'_f\}$ de feuilles de $\mathcal{T}_R$, copier $Z_f \cap Z'_f$ dans tout sac $Z \in \mathcal{T}_R[Z_f, Z'_f]$.

Montrons que DecomposerIntersection($\mathcal{T}, R$) remplace R dans $\mathcal{T}$ par un arbre $\mathcal{T}_0$ de décomposition arborescente de R . En effet, selon que les chemins dans R sont des sillons ou pas, on a soit un seul pôle $r(X)$, soit deux pôles $r(X)$ et $r(Y)$. Pour chaque sac Z adjacent à R , l'étape 1 identifie l'ensemble des sillons ou chemins de R couvrant $R \cap Z$ et les met dans un sac noté R_Z . On note $\mathcal{S}$ l'ensemble des sacs engendrés par $N(R)$ et $R_0 = R \setminus \mathcal{S}$. Ensuite, on identifie les sacs de $\mathcal{S}$ qui

contiennent des sillons s tel que $N(s) \cap R_0 \neq \emptyset$ et on met $s \cup \{N(s) \cap R_0\}$ dans un sac noté R_s et on ajoute R_s à $\mathcal{S}$. Puis, on attache les sacs de $\mathcal{S}$ à $\mathcal{T}$ comme suit : pour chaque sac $R_z \in \mathcal{S}$ engendré par $Z \in N(R)$, si $Z \neq \text{père}(R)$, faire $\text{père}(Z) = R_Z$. Donc, pour chaque sac $Z \in N(R)$ cette opération construit un nouvel arbre $\mathcal{T}_{R_Z}$ ayant comme racine R_Z et comme fils Z . En outre, on pose $P = \text{père}(R)$, et R_P le sac de $\mathcal{S}$ contenant $R \cap P$. Puis on remplace le sac R dans $\mathcal{T}$ par R_P . On insère ainsi le sac R_P dans $\mathcal{T}$. Il reste à rattacher les sous-arbres $\mathcal{T}_{R_Z}$ engendrés par chaque sac $Z \in N(R)$ pour obtenir le résultat escompté. Ce rattachement vas se faire dans la boucle “TANT QUE” de l’étape 2.

En effet, après avoir marqué R_P comme une feuille de $\mathcal{T}$, on passe à l’étape 2. Dans cette étape, on vérifie si la seule feuille R_P de $\mathcal{T}$ intersecte un sac $Z \in \mathcal{S}$ en au moins un sommet qui n’est pas un pôle dans R . Si tel est le cas, pour tout $Z \in \mathcal{S}$ intersectant R_P en au moins un sommet qui n’est pas un pôle et tel que $\text{père}(Z) = \emptyset$, on définit $\text{père}(Z) = R_P$ et Z devient feuille. Ensuite, si tous les sacs de $\mathcal{S}$ qui intersectent R_P sont rattachés, le sac R_P devient simple. On réitère ce procédé jusqu’à ce que tous les sacs de $\mathcal{S}$ obtiennent tous un père dans $\mathcal{T}$. Ainsi, on obtient un arbre $\mathcal{T}$ où tous les sacs issus de la décomposition de R sont attachés.

S’il existe un sac $Z \in \mathcal{S}$ tel que $\text{père}(Z) = \emptyset$ et qu’aucune feuille de $\mathcal{T}_R$ n’intersecte aucun sac Z de $\mathcal{S}$ tel que $\text{père}(Z) = \emptyset$, alors c’est que les seuls sommets intersectant Z avec les sacs déjà rattachés dans $\mathcal{T}_R$ sont les pôles de R . Dans ce cas, on choisit arbitrairement un sac $Z \in \mathcal{S}$ tel que $\text{père}(Z) = \emptyset$, et on le rattache à R_P et puis on le marque comme feuille grâce à l’étape 2.2. Ainsi le procédé reprendra à la prochaine itération et on assure le fait que tous les sacs de $\mathcal{S}$ soient rattachés dans $\mathcal{T}_R$.

Nous avons donc construit un arbre de décomposition dans lequel tous les sacs de $\mathcal{S}$ sont rattachés. De plus, $\mathcal{S}$ couvre tous les sommets de R , donc tous les sommets de R sont couverts par $\mathcal{T}$. En outre, on sait que toute arête e dont les deux extrémités sont dans R est aussi telle que les deux extrémités appartiennent à un sillon couvert par R . Or, on sait par définition que R ne contient que des sillons couvrant l’intersection entre deux sacs de $N(R)$. Alors pour tout sillon s de R tel que $e \in s$, il existe un sac $Z \in N(R)$ tel que $s \cap \{Z \cap R\} \neq \emptyset$. D’où il existe un sac $W_Z \in \mathcal{S}$ engendré par Z et $s \subseteq W_Z$. Donc W_Z contient les extrémités de l’arête e . Nous avons donc montré que $\mathcal{T}_R$ vérifie bien les conditions (i) et (ii) de la définition 2.1.

Il reste donc à vérifier que la condition (iii) de la définition 2.1 est respectée par $\mathcal{T}$ après le traitement du sac R . En effet, l’étape 3 assure cette propriété à toutes les feuilles à la sortie de l’étape 2. Donc, s’il existe deux sacs X et Y de $\mathcal{T}$ qui ne vérifient pas la condition (iii), alors ni X ni Y est une feuille. Par ailleurs, par définition de X et Y , il existe deux sacs $Z_X \in N(R)$ et $Z_Y \in N(R)$ tels que $Z_X \cap R \subseteq X$ et $Z_Y \cap R \subseteq Y$. Si X et Y partagent un sillon s de diamant, alors s est dans $Z_X \cap R$ et $Z_Y \cap R$. Le sillon s appartient à la fois au diamant contenu dans Z_X et à celui contenu dans Z_Y . Or, nous savons grâce à la propriété 2.8 que l’intersection entre deux diamants est dans au plus deux sillons. Donc, X et Y partagent au plus deux sillons et tout sillon donné de R appartient à au plus deux sacs de $\mathcal{S}$. Posons $W \in \mathcal{T}[X, Y]$ tel que $X \cap Y \not\subseteq W$; W existe car X et Y ne vérifient pas la condition (iii).

Soit s un sillon qui appartient à la fois à X et à Y . Soit $X_1 \in \mathcal{T}[X, Y]$ et X_1

adjacent à X . On a donc $X \cap X_1 = s' \neq s$ car, d'après la propriété 2.8, s ne peut pas être dans plus de deux sacs de $\mathcal{S}$. Vu que X n'est pas une feuille, alors il existe un sac X_2 adjacent à X et $X_2 \notin \mathcal{T}[X, Y]$. On a donc $X \cap X_2 = s'' \neq s'$ car on a $s' \in X \cap X_1$ et $s'' \neq s$. Donc X contient au moins 3 sillons ; ce qui est une contradiction avec l'hypothèse que chaque sac de $\mathcal{S}$ possède au plus deux sillons de R . Donc W n'existe pas et donc $\mathcal{T}$ vérifie la condition (iii) de la définition 2.1.

Traisons maintenant le cas où X et Y ne partagent que des demi-sillons. Les sacs X et Y contiennent alors chacun les intersections de composantes connexes C_X, C_Y issues d'un demi-diamant. Posons C_X la composante connexe telle que $X \setminus r(X) \subseteq C_X$ et C_Y la composante connexe telle que $Y \setminus r(X) \subseteq C_Y$.

Supposons que $W \in \mathcal{T}[X, Y]$ tel que $X \cap Y \not\subseteq W$. Par définition, on a C_X et C_Y qui se déconnectent lorsqu'on supprime $r(X)$ du demi-diamant. Donc, vu que par définition $X \subseteq C_X \cup r(X)$ et $Y \subseteq C_Y \cup r(X)$, alors $X \cap Y = r(X)$ et donc, $X \cap Y \subseteq W$ car tous les sacs engendrés par R contiennent $r(X)$. Ce qui est une contradiction avec l'hypothèse. D'où W n'existe pas et donc $\mathcal{T}$ vérifie bien la condition (iii) de la définition 2.1.

Montrons maintenant comment décomposer tout sac contenant un demi-diamant. La procédure **DECOMPOSER** donne une décomposition arborescente pour tout graphe donné avec un arbre couvrant. Elle sera appelée sur toute composante connexe issue d'un demi-diamant. En effet, pour chaque demi-diamant $\Delta(X)$ de pôle $r(X)$, chaque composante connexe de $G[\Delta(X)] \setminus r(X)$ est couverte par un sous arbre de T_X . Donc, vu que le sac Z contenant la composante $G[\Delta(X)] \setminus r(X)$ dans $\mathcal{T}$ contient $r(X)$, alors il existe un sous-arbre de T_X qui couvre Z . Donc, décomposer Z revient à décomposer un sous-graphe $G[Z]$ de G ; il admet un arbre couvrant, T_X et il est planaire. Grâce à la propriété 2.6, $G[Z]$ possède une décomposition arborescente dans laquelle chaque sac contient au plus 3 chemins de T_X .

Principe de la procédure DECOMPOSER : Considérons le sous-graphe H de $G[Z]$ tel que chaque couple (s_1, s_2) de sillons de Z appartenant à un même diamant est remplacé par le sillon $\max(s_1, s_2)$ qui est marqué de tel sorte qu'on puisse l'identifier. Alors H est planaire et admet un arbre T_X couvrant de profondeur $3\ell/2$. On peut donc supposer l'existence d'un arbre $\mathcal{T}_0$ de décomposition de H , tel que chaque sac contient au plus 3 chemins de T_X de longueur $\leq 3\ell/2$. Pour chaque sac X de $\mathcal{T}_0$ contenant un sillon s marqué, dupliquer s dans X . Dans le pire des cas X possède 3 sillons marqués et donc dans le nouvel arbre $\mathcal{T}_0$ produit on a au plus 6 sillons et donc 6 chemins de longueur au plus $3\ell/2$. Pour chaque sac W adjacent à Z dans $\mathcal{T}$, chercher dans $\mathcal{T}_0$ le premier sac $Z_0 \in \mathcal{T}_0$ partant de la racine de $\mathcal{T}_0$ tel que $W \cap Z_0 \neq \emptyset$ et faire : si $\text{père}(W) = Z$, alors $\text{père}(W) = Z_0$, sinon $\text{père}(Z_0) = W$. En effet, chaque sac W adjacent à Z est un diamant et donc d'après la propriété 2.6 W intersecte toute composante connexe de Z en au plus deux sillons.

Montrons que l'arbre $\mathcal{T}$ ainsi obtenu vérifie bien les conditions de la définition 2.1 : Nous savons que $\mathcal{T}_0$ respecte les conditions de la définition 2.1. Nous savons aussi que l'arbre $\mathcal{T}_1$ obtenu en contractant dans $\mathcal{T}$ les arêtes de $\mathcal{T}_0$ en un sac vérifie aussi la définition 2.1. Ce qui reste à vérifier c'est de savoir si la condition

(iii) reste vérifiée après avoir connecté $\mathcal{T}_0$ à $\mathcal{T}$. En effet, supposons que la condition (iii) n'est plus respectée par $\mathcal{T}$ après la connexion de $\mathcal{T}_0$. Alors, il existe X_0, X_1 et X_3 des sacs de $\mathcal{T}$ tel que $X_1 \in \mathcal{T}[X_0, X_2]$ et $X_0 \cap X_2 \not\subseteq X_1$. Nous pouvons facilement remarquer que X_0 et X_2 ne sont pas couverts tous les deux par $\mathcal{T}_0$ car sinon $\mathcal{T}_0$ ne vérifierait pas la condition (iii). Par ailleurs, nous pouvons aussi montrer facilement qu'au moins X_0 ou X_2 est couvert par $\mathcal{T}_0$. En effet, supposons que $X_0 \notin \mathcal{T}_0$ ou $X_2 \notin \mathcal{T}_0$. Dans ce cas, $X_1 \in \mathcal{T}_0$ car sinon l'arbre $\mathcal{T}$ obtenu en contractant l'arbre $\mathcal{T}_0$ ne vérifierait pas la condition (iii). Donc le sac Z obtenu en contractant $\mathcal{T}_0$ contient $X_0 \cap X_2$. Posons Z_0 le premier sac de $\mathcal{T}_0$ qui intersecte X_0 et Z_1 le premier sac de $\mathcal{T}_0$ qui intersecte X_2 . On a donc $Z_1 \neq Z_0$. Or, par construction de $\mathcal{T}_0$, tout sac de $\mathcal{T}_0$ qui intersecte un sac W de $\mathcal{T}_1$ contient $W \cap \mathcal{T}_0$. Ce qui deviendrait faux si $Z_1 \neq Z_0$. Donc, soit X_0 , soit X_2 est couvert par $\mathcal{T}_0$. Supposons sans perdre de généralité que $X_2 \in \mathcal{T}_0$. Donc, si $X_1 \in \mathcal{T}_0$, alors $\mathcal{T}_1$ ne vérifierait pas la condition (iii); ce qui est une contradiction avec l'hypothèse. Sinon, si $X_1 \notin \mathcal{T}_0$, alors le sac $Z \in \mathcal{T}_1$ résultant de la contraction de $\mathcal{T}_0$ serait tel que $X_1 \in \mathcal{T}_1[Z, X_2]$ et $Z \cap X_2 \not\subseteq X_1$. Donc $\mathcal{T}_1$ ne vérifie pas la condition (iii); ce qui est une contradiction. Par conséquent, $\mathcal{T}_1$ vérifie bien la condition (iii) de la définition 2.1.

Nous avons donc montré que suivant les 4 types de contenu qu'un sac $X \in \mathcal{T}_0$ peut contenir, la procédure **MAIN** produit en temps polynomial en n un arbre de décomposition dans lequel le contenu de X est dispatché dans plusieurs sacs de $\mathcal{T}$ et dont chacun contient au plus 6 chemins de longueur au plus $3\ell/2$. De plus, chaque chemin étant composé d'au plus deux plus courts chemins, chaque sac issu de X possède donc au plus 12 plus courts chemins de longueur au plus ℓ .

Par ailleurs, vu que la procédure traite récursivement les sacs de $\mathcal{T}_0$, alors à la terminaison de la procédure, chaque sac de $\mathcal{T}$ possède au plus 12 plus courts chemins de longueur au plus ℓ . Nous avons donc montré que la procédure **MAIN** produit en temps polynomial, un arbre de décomposition de G dans lequel chaque sac possède au plus 12 plus courts chemins de longueur au plus ℓ . Donc, nous avons montré le théorème 2.9. $\square$

2.5 Conclusion

Le résultat principal de ce chapitre est un algorithme polynomial qui pour tout graphe planaire de longueur arborescente ℓ , donne une décomposition arborescente $\mathcal{T}$ dans laquelle chaque sac a au plus 12 plus courts chemins de longueur ℓ . En effet, chaque sac possède au plus 6 chemins de longueur $\leq 3\ell/2$, composés chacun d'au plus deux plus courts chemins. Cette décomposition met donc en évidence une relation entre la largeur arborescente et la longueur arborescentes de tout graphe planaire.

Une amélioration de notre résultat serait de montrer que tout graphe de genre borné et de longueur arborescente ℓ admet une décomposition arborescente dans laquelle chaque sac contient un nombre fini de plus courts chemins de longueur au plus $O(\ell)$.

Chapitre 3

La structure des graphes sans mineur H

3.1 Introduction

Dans ce chapitre, nous nous intéressons à l'étude de la structure des graphes qui excluent un certain graphe H comme mineur.

Depuis la classification par Wagner des graphes sans mineur K_5 [Wag37], l'étude des graphes contenant (ou excluant) un certain graphe comme mineur ne cesse de passionner les chercheurs.

Un des résultats les plus connus sur les mineurs de graphe est la reformulation par Wagner [Wag37] du théorème de Kuratowski disant qu'un graphe G est planaire si et seulement si G est sans mineur K_5 et $K_{3,3}$. De même, dans ses travaux Wagner montre que tout graphe sans mineur $K_{3,3}$ admet une décomposition arborescente $\mathcal{T}$ dans laquelle les sacs de $\mathcal{T}$ s'intersectent en au plus 2 sommets et chaque sac $X \in \mathcal{T}$ induit un sous-graphe de soit K_5 soit d'un graphe planaire. Il montre aussi que tout graphe sans mineur K_5 admet une décomposition arborescente $\mathcal{T}$ dans laquelle les sacs de $\mathcal{T}$ s'intersectent en au plus 3 sommets et chaque sac $X \in \mathcal{T}$ induit soit un graphe planaire soit le graphe de Wagner V_8 de la figure 3.1.

La connaissance de la structure d'un graphe facilite dans certains cas la résolution de problèmes complexes. Par exemple, l'étude de la structure des graphes sans mineur K_{r+1} a permis à Robertson et *al.* [RST93] de prouver pour $1 \leq r \leq 5$ le problème de la r -coloration des graphes sans mineur K_{r+1} : la conjecture d'Hadwiger [Had43]. Dans cet article, ils montrent aussi que la 5-coloration des graphes sans mineur K_6 est équivalente au théorème des 4 couleurs des graphes planaires. Ils montrent que tout contre exemple minimal est un graphe planaire plus un sommet.

En outre dans [RS03], Robertson et Seymour montrent que pour tout graphe H , il existe une constante k_H telle que tout graphe G sans mineur H possède une décomposition arborescente $\mathcal{T}$ dans laquelle les sacs de $\mathcal{T}$ s'intersectent en au plus k_H sommets et, tout sac $X \in \mathcal{T}$ tel que $|X| > k_H$ induit un sous-graphe " k_H -almost embeddable" sur une surface Σ dans laquelle H n'est pas dessinable. En gros, un graphe G est " h -almost embeddable" sur une surface Σ s'il existe un ensemble Y d'au plus h sommets appelés des "*apex*" tels que $G \setminus Y$ peut être obtenu à partir d'un graphe G_0 plongé sur Σ en attachant au plus h graphes de "pathwidth" au plus h à l'intérieur de h faces de G_0 de manière ordonnée. Ce résultat est une généralisation

de la caractérisation des graphes sans mineur $K_{3,3}$ et K_5 [Wag37]. Ainsi, la surface Σ considérée est la sphère. Cependant, la constante k_H est gigantesque et aucune majoration sur k_{K_5} n'est connue.

De nombreuses caractérisations sont connues lorsque le mineur H exclu contient au plus 5 sommets, en particulier pour $H = K_5$ [Wag37]. Le problème est beaucoup plus difficile lorsque H contient 6 sommets, voir plus. Par exemple, la caractérisation des graphes sans mineur K_6 est non résolue, et elle est en relation avec la conjecture d'Hadwiger concernant le nombre chromatique des graphes sans mineur K_r (voir [Kaw07] pour une plus amples discussions sur l'importance des graphes sans mineur K_6).

Il existe plusieurs structures bien connues qui garantissent l'existence de certains mineurs dans un graphe G si ce dernier est assez grand. Par exemple, tout graphe 5-connexe ayant au moins 11 sommets contient un 3-cube comme mineur [Mah00]. Oporowski, Oxley et Thomas prouvent que tout graphe 4-connexe, suffisamment grand possède un mineur appartenant à la famille des graphes 4-connexes [OOT93]. Ils ont aussi donné un résultat similaire pour les grands graphes 3-connexes. Récemment, Kawarabayashi et Maharry ont montré un résultat similaire dans des travaux en prépublication. Citons également la conjecture ci-dessous de Jørgensen :

Conjecture 3.1 [Jør94] *Tout graphe 6-connexe G sans mineur K_6 possède un sommet u tel que $G \setminus \{u\}$ est planaire.*

Cette conjecture est toujours ouverte même si des travaux en prépublication de DeVos, Kawarabayashi, Norine, et Thomas ont permis de prouver qu'elle est vraie pour des graphes de grande taille. Par ailleurs, dans [Kaw07], Kawarabayashi étudie la structure des graphes sans mineur K_6 et K_7 . De même, dans [Ch10] on donne une caractérisation des graphes sans mineur K_5 . Dans [CRS08], les travaux de Chudnovsky *et al.* ont permis de montrer que les graphes à n sommets sans mineur $K_{2,r}$ ont au plus $(n-1)(r+1)/2$ arêtes. La largeur arborescente des graphes planaires sans mineur $K_{2,r}$ est donnée par Thilikos [Thi99] et elle est $r+2$. Pour le cas général, la borne est $2r-2$ [BLTT97].

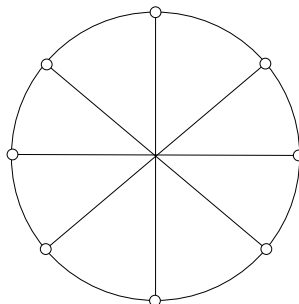


FIG. 3.1: W_8 .

3.2 Présentation des résultats

La première partie de ce chapitre sera consacré essentiellement à la démonstration du théorème suivant :

Théorème 3.2 *Pour tout graphe 2-connexe G à n sommets, en temps $O(n)$ on peut soit extraire un mineur $K_{2,4}$ soit extraire deux sommets u, v (un sommet si G est planaire) tels que $G \setminus \{u, v\}$ est planaire-extérieur.*

Un corollaire immédiat du théorème 3.2 est que les graphes sans mineur $K_{2,4}$ sont de largeur arborescente au plus 4 dans le cas général, et au plus 3 dans le cas des graphes planaires. En effet, un mineur $K_{2,4}$ de G peut entièrement être dans une composante 2-connexe de G , et la largeur arborescente de G est le maximum sur les largeurs arborescentes de ses composantes 2-connexes.

Ces deux bornes sont optimales à cause de K_5 et K_4 pour les graphes planaires. La meilleure borne connue était 6 pour le cas général, et elle est dérivée de la borne supérieure pour la largeur arborescente des graphes sans mineur $K_{2,r}$ qui est de $2r - 2$ (cf. Bodlaender *et al.* [BKK95a]). Pour les graphes planaires, la meilleure borne connue était 4 et plus généralement $r + 2$ pour les graphes planaires sans mineur $K_{2,r}$ [Thi99].

Une autre conséquence de ce résultat est que les graphes sans mineur $K_{2,4}$ sont 5-colorables. En effet, il est bien connu que les graphes planaire-extérieurs sont 3-colorables. Par conséquent, en colorant le sous-graphe $G \setminus \{u, v\}$ avec 3 couleurs, on peut étendre cette coloration à toute le graphe en ajoutant au plus 2 couleurs pour colorier proprement les deux sommets u et v . De plus ce résultat est optimal à cause de K_5 .

Nous nous sommes naturellement posé la question de savoir si le théorème 3.2 pouvait être généralisé aux graphes sans mineur $K_{2,r}$ pour $r > 4$. Malheureusement, comme nous le verrons dans la section 4.3, nous montrons que pour tout entier k , il existe un graphe G_k 2-connexe sans mineur $K_{2,5}$ pour lequel il est nécessaire d'enlever au moins k sommets pour le rendre planaire-extérieur. Par exemple le graphe de la figure 3.2 est sans mineur $K_{2,5}$ et pour le rendre planaire-extérieur, il faut supprimer au moins 4 sommets. Ainsi, la seule possibilité qu'on a pour étendre ce théorème à $r > 4$ est de supposer que le graphe est fortement connexe.

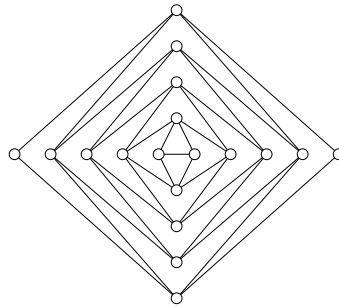


FIG. 3.2: Un graphe G sans mineur $K_{2,5}$.

Nous nous sommes également intéressés dans la section 3.4 à l'étude de la borne supérieure de la largeur arborescente des graphes planaires. Une des propriétés les plus intrigantes de la théorie des mineurs de graphe est celle qui permet de borner la largeur arborescente des graphes par une fonction dépendant du mineur qu'il exclut. Dans [RST94], Robertson *et al.* ont montré que la largeur arborescente d'un graphe est bornée si celui-ci exclut un graphe planaire. Nous nous sommes intéressés à cette question pour les graphes planaires. Il est connu que la largeur arborescente des graphes planaires excluant un graphe planaire à r sommets comme mineur est $O(r)$ [RST94].

Cette borne ne peut pas être améliorée dans le cas général, car il existe des graphes planaires ayant $O(r)$ sommets et qui sont exclus par la grille $r \times r$ qui est de largeur arborescente r . Cependant, nous pouvons l'améliorer si nous restreignons la classe des mineurs à exclure.

Plus précisément, nous nous intéressons à une famille $\mathcal{F}$ de graphes et une fonction f telle que la largeur arborescente de tout graphe planaire excluant un graphe d'ordre r de $\mathcal{F}$ comme mineur est au plus $f(r)$. En particulier, nous allons déterminer une grande famille $\mathcal{F}$ de graphes telles que $f(r) = O(\sqrt{r})$, appelé la famille des graphes possédant un dessin sur une grille $O(1) \times r$.

De manière plus précise, nous montrons le théorème suivant :

Théorème 3.3 *Si H est un graphe dessinable sur une grille $p \times q$ (tel que les arêtes de H sont de segments de droite), alors pour tout graphe G planaire sans mineur H , on a : $\text{tw}(G) = O(p^{3/2}\sqrt{q})$.*

3.3 La structure des graphes sans mineur $K_{2,4}$

La démonstration du théorème 3.2 est une combinaison des deux résultats suivants :

Lemme 3.4 *Si G est un graphe planaire, 2-connexe, alors en temps linéaire on peut soit extraire mineur $K_{2,4}$ soit extraire un sommet u tel que $G \setminus \{u\}$ est planaire-extérieur.*

Lemme 3.5 *Si G est un graphe non-planaire, 2-connexe à n sommets, alors en temps $O(n)$ on peut extraire soit un mineur $K_{2,4}$ soit un sommet u tel que $G \setminus \{u\}$ est planaire.*

3.3.1 Préliminaires

Avant de passer à la démonstration du lemme 3.4, nous avons besoin de définir quelques éléments que nous aurons à manipuler tout au long de la preuve.

Soient M un chemin et C un cycle dans un graphe G planaire donné.

On note par $M[u, v]$ un chemin de u à v dans M et $C^+[u, v]$ un chemin de u à v dans C , suivant un parcours de C dans le sens des aiguilles d'une montre. Il est possible que le chemin passe deux fois par un même sommet.

On note par $M]u, v[$ le sous-chemin $M[u, v] \setminus \{u, v\}$.

Si G plongé sur le plan, le cycle C partage le plan en deux régions connexes dont l'une des deux régions, notée R est la région intérieur du plan délimité par le cycle C ; elle est homomorphe à un disque ouvert. On note par $\text{IN}(C)$ l'ensemble des sommets de G appartenant à la région R . Notons que $\text{IN}(G)$ n'est pas forcément connexe même si G est connexe.

Pour toute composante connexe X de $G \setminus C$, un attachement de X est un sommet de C qui est adjacent à X dans G .

Pour tout graphe G et tout mineur $K_{2,r}$ de G , on note A et B les deux composantes connexes de G qui constituent les deux pôles du mineur, c'est-à-dire les deux sommets de degré r . Dans ce cas, il existe r chemins disjoints de longueur au moins deux entre A et B .

La longueur d'un chemin P notée $|P|$ est le nombre d'arêtes qui composent P . Pour basculer une composante connexe X attachée au cycle C à l'extérieur de la région R , il suffit de procéder à un transfert symétrique de X par rapport à C de la région R vers la région R' symétrique de R par rapport à C . Deux régions symétriques par rapport à C partagent au moins une arête de C .

3.3.2 Démonstration du lemme 3.4 : le cas planaire

Nous avons besoin de la définition suivante pour la démonstration :

Définition 3.6 (plongement-LMR) *Un plongement-LMR d'un graphe planaire G est un plongement de G avec trois chemins distingués de G , notés L, M et R , partageant uniquement leurs extrémités et tel que :*

1. $L \cup R$ est le bord de la face extérieure de G ;
2. $\text{IN}(L \cup M)$ et $\text{IN}(M \cup R)$ ne contiennent aucun sommet ;
3. $\text{IN}(M \cup R)$ ne contient aucune arête dont les deux extrémités sont dans M ;
4. M est de longueur au moins deux.

Les propriétés 1 et 2 du plongement-LMR impliquent que les chemins L, M, R couvrent les sommets du graphe. La figure 3.3 en est une illustration.

La démonstration du théorème 3.2 se résume à la démonstration des deux lemmes suivants.

Lemme 3.7 *Soit un plongement-LMR d'un graphe planaire G . Il existe un algorithme qui en temps linéaire soit extrait un mineur $K_{2,4}$, soit trouve un sommet x tel que $G \setminus \{x\}$ est planaire-extérieur.*

Démonstration : Soit u, v les deux sommets commun aux chemins L, M, R . On peut remarquer que par définition du plongement-LMR, on a $G \setminus R]u, v[$ est un sous-graphe planaire-extérieur. Pour démontrer le lemme, nous allons appliquer tant que c'est possible les règles suivantes :

Règle 1 : Si $\text{IN}(M \cup R)$ ne contient aucune arête reliant un sommet de M et un sommet de R , alors en supprimant u la face extérieure du sous-graphe résultant couvre tous les sommets de M . Or cette même face couvrirait déjà les sommets de $L \cup R$. Donc, le plongement résultant de $G \setminus \{u\}$ est planaire-extérieur.

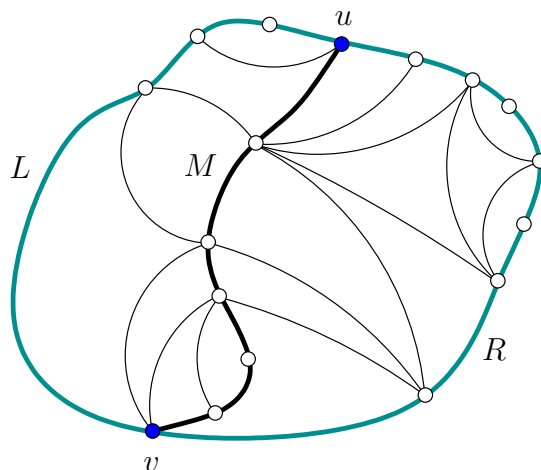


FIG. 3.3: Un plongement-LMR.

Règle 2 : S'il existe un sommet w de R incident à toute arête de $\text{IN}(M \cup R)$ ayant une extrémité dans M , alors on supprime w et le plongement résultant $G \setminus \{w\}$ est planaire-extérieur.

Règle 3 :

1. S'il existe un sommet w de M qui est adjacent à u , et à un sommet x de R . Alors, on construit un nouveau plongement-LMR en mettant à jour les chemins L, M, R de la manière suivante :
 - L devient $L \cup R[u, x]$;
 - M devient $M[v, w] \cup \{wx\}$; et
 - R devient $R[x, v]$.
2. Sinon, s'il existe un sommet w de M qui est adjacent à v , et à un sommet x de R . Alors, on construit un nouveau plongement-LMR en mettant à jour les chemins L, M, R de la manière suivante :
 - L devient $L \cup R[v, x]$;
 - M devient $M[u, w] \cup \{wx\}$; et
 - R devient $R[x, u]$.

On remarque que dans tous les cas (1 ou 2) de cette règle, le nombre de sommets couverts par R diminue d'au moins un. Ce qui veut dire qu'on va converger vers une configuration dans laquelle l'une des règles 1 ou 2 peut s'appliquer.

Si aucune des règles 1, 2 ou 3 ne peut s'appliquer, alors G contient un mineur $K_{2,4}$. En effet, la règle 3 ne peut pas s'appliquer, car u a un voisin w_u dans M qui n'admet pas de voisin dans R . C'est aussi valable pour le sommet v qui a un voisin w_v dans M et w_u n'admet aucun voisin dans R . Grâce à la règle 1, $M[w_u, w_v]$ contient au moins un sommet qui a un voisin dans R . Grâce à la règle 2, toutes les arêtes entre M (et partant de $M[w_u, w_v]$) et R ne sont pas incidentes à un seul sommet de R . Donc, il y a deux sommets différents x_u et x_v dans R possédant un

voisin dans $M]w_u, w_v[$.

Par conséquent, on peut construire un mineur $K_{2,4}$ en définissant $A = M]w_u, w_v[$ et $B = L[u, v]$, comme le montre la figure 3.4.

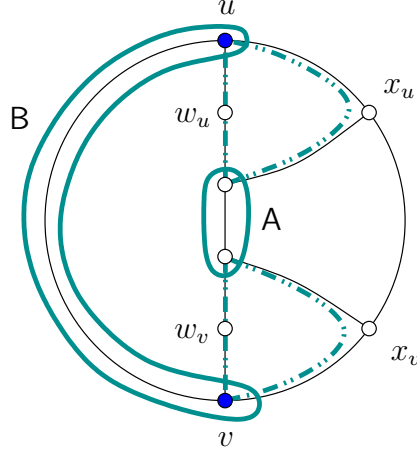


FIG. 3.4: Un mineur $K_{2,4}$ pour la preuve du lemme 3.7.

Si nous ne sommes pas dans le cas où nous pouvons construire un mineur $K_{2,4}$, alors nous pouvons terminer par la règle 1 ou la règle 2 et on cherche un sommet tel que sa suppression rend le plongement planaire-extérieur.

Pour conclure, nous observons que cela peut se faire en temps linéaire en commençant par appliquer tant que possible la règle 3. En effet, on aura à visiter chaque arête du graphe un nombre constant de fois. L'application de la règle 1 et la règle 2 se fait en temps linéaire. $\square$

Lemme 3.8 *Tout graphe planaire 2-connexe qui n'est pas un cycle soit admet un plongement-LMR soit contient un mineur $K_{2,4}$. De plus, un tel plongement ou un tel mineur peut être construit en temps linéaire.*

Démonstration : Soit G un graphe planaire 2-connexe différent d'un cycle. Étape après étape, nous allons construire le plongement-LMR pour G et dans le cas où ce plongement n'est pas possible, nous allons extraire un $K_{2,4}$.

Considérons un plongement arbitraire de G dans le plan. Ce plongement peut être obtenu en temps linéaire. Soit C le bord de la face extérieure de ce plongement. C est un cycle simple car sinon G ne serait pas 2-connexe. Si $G \setminus C$ est vide, alors G est planaire-extérieur. Dans ce cas, pour obtenir un plongement-LMR il suffit juste de considérer une arête uv de $\text{In}(C)$ et de la basculer vers l'extérieur de $\text{In}(C)$. Une telle arête existe car G n'est pas un cycle. On pose $L = C^+[v, u]$, $M = C^+[u, v]$ et $R = uv$ et on obtient un plongement-LMR. Où $|M| \geq 2$ car il n'y a pas d'arêtes multiples.

Supposons maintenant que G est planaire et non planaire-extérieur, c'est-à-dire $G \setminus C$ est non vide. Soit X une composante connexe de $G \setminus C$. Nous allons décrire comment X est attaché à C . Nous utiliserons cette construction plus tard dans la

preuve et on supposera toujours C comme étant un cycle du graphe G . On s'appuiera de la figure 3.5.

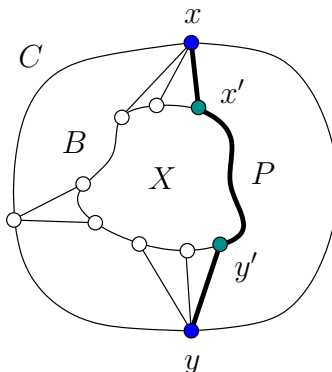


FIG. 3.5: Les attachements de X à C .

Soit x, y deux attachements de X tels que x et y sont rencontrés consécutivement en parcourant C dans le sens de aiguilles d'une montre. Soit B le bord de la face extérieure de X (dans le plongement actuel de G). Notons que B n'est pas nécessairement un cycle, il peut contenir des séparateurs de X car X n'est pas forcément 2-connexe. Soit x' (respectivement y') le dernier voisin de x (respectivement le premier voisin de y) rencontré en parcourant B dans le sens de aiguilles d'une montre (il est possible d'avoir $x' = y'$). On remarque que $B^+[x', y']$ est un chemin simple car il ne peut pas contenir deux fois le même sommet. En effet, si ce n'était pas le cas, alors par le fait de la planarité et la 2-connectivité de G , on aurait une composante 2-connexe de X qui aurait un voisin dans $C^+[x, y]$, ce qui est une contradiction car x et y sont des attachements consécutifs de X . Par conséquent, $P = \{xx'\} \cup B^+[x', y'] \cup \{y'y\}$ est un chemin simple de longueur au moins deux.

On supposera que X et les attachements x, y sont choisis tels qu'il n'existe aucune composante connexe de $G \setminus C$ dans la région $\text{IN}(C^+[x, y] \cup P)$. Cette région ne contient aucun sommet de G mais peut contenir des arêtes dont les deux extrémités sont tous dans $C^+[x, y]$. Les sous-graphes X et P peuvent être déterminés en temps linéaire.

On pose $L = C^+[y, x]$, $M = P$, et $R = C^+[x, y]$. Il est facile de voir que les chemins L, M, R construits partagent uniquement leurs extrémités et ils ont les propriétés suivantes :

1. $L \cup R$ est le bord de la face extérieure de G ;
2. $\text{IN}(M \cup R)$ ne contient aucun sommet ;
3. $\text{IN}(M \cup R)$ ne contient aucune arête ayant ses deux extrémités dans M ;
4. M est de longueur ≥ 2 .

En d'autres termes, le plongement de G diffère d'un plongement-LMR par le fait que $\text{IN}(L \cup M)$ peut contenir des sommets. On note par u, v les sommets communs à la fois aux chemins L, M, R , c'est-à-dire, on pose $u = x$ et $v = y$.

Maintenant, on met à jour le plongement en appliquant quelques règles de modification tant que les 4 propriétés précédentes sont vérifiées. Pour montrer que ce

processus converge vers un plongement-LMR, nous avons besoin d'introduire la fonction de potentiel $\phi = \phi(L, M, R)$ définie comme suit :

$$\phi := |V(L \cup M \cup R)| + n \cdot \min \{|R| - 1, 1\}$$

Où n représente le nombre de sommets du graphe. En d'autres termes, si R est une arête ($|R| = 1$), alors ϕ est le nombre de sommets couverts par les trois chemins. Sinon, si $|R| > 1$, on ajoute n à cette somme. Notons que si $|R| > 1$, alors pour accroître ϕ on a besoin d'augmenter le nombre de sommets couverts et en même temps de maintenir $|R| \geq 2$. Initialement, on a $\phi \geq 4$, et $\phi \leq 2n$ pour tout choix de L, M, R . Nous allons montrer que ϕ augmente d'au moins 1 à chaque application d'une des règles sauf à la dernière application de la règle 1. Cependant, il y a une manière d'appliquer les règles pour que chaque application ait un amortissement en temps de $O(1)$ et pour que la complexité au total soit en $O(n)$.

Règle 1 : Il n'y a aucun sommet dans $\text{IN}(L \cup M)$. Dans ce cas, le plongement courant est un plongement-LMR.

Si on suppose que la règle 1 n'est pas vérifiée, on pose X une composante connexe de $G \setminus (L \cup M \cup R)$ et A l'ensemble des attachements de X . Grâce aux 4 propriétés précédentes, X est plongé dans la région $\text{IN}(L \cup M)$.

Règle 2 : Les attachements de X induisent exactement une arête de L , ou une arête de M . Alors, on bascule la composante X vers l'extérieur de $\text{IN}(L \cup M)$, et on met à jour convenablement L ou M . Le nouveau plongement couvre au moins un sommet de plus (un sommet de X), et donc ϕ croît strictement.

On pose x, y deux attachements de X au cycle $C = L \cup M$, et on définit le chemin P comme défini précédemment (cf. figure 3.5). On suppose que la région $\text{IN}(P \cup C[x, y])$ est vide ; cela est possible avec un choix approprié de X . On a l'une des configurations suivantes :

- A. $\{x, y\} \subseteq L$;
- B. $\{x, y\} \not\subseteq L$ et $\{x, y\} \not\subseteq M$
- C. $\{x, y\} \subseteq M$ et $\{x, y\} \neq \{u, v\}$

Pour le cas A, on suppose qu'on rencontre les sommets x, u, v, y dans cet ordre lorsque l'on parcourt C dans le sens des aiguilles d'une montre, avec la possibilité de sélectionner une autre composante X et en échangeant le rôle de x et y (cf. figure 3.6 (a)). Posons $M' = L[v, y] \cup P \cup L[x, u]$. Notons que par le choix de x, y , la région $\text{IN}(M' \cup M[u, v])$ ne contient aucun sommet car sinon le cas A ne serait pas vérifié.

Règle 3 : Si le cas A est vérifié et R est une arête, alors on bascule l'arête R vers la gauche, c'est-à-dire passer de (L, M, R) vers (R, M', M) . Les quatre propriétés sont toujours vérifiées et le nouveau chemin R (précédent chemin M) est de longueur ≥ 2 , et donc ϕ croît strictement. Par exemple, la figure 3.7 (a) est transformée en la figure 3.7 (b) : l'arête R est basculée vers la gauche et forme ainsi le nouveau chemin L ; le chemin M' en rouge deviendra le chemin M et l'ancien chemin M devient le nouveau chemin R . On a donc la longueur du nouveau chemin R qui est ≥ 2 .

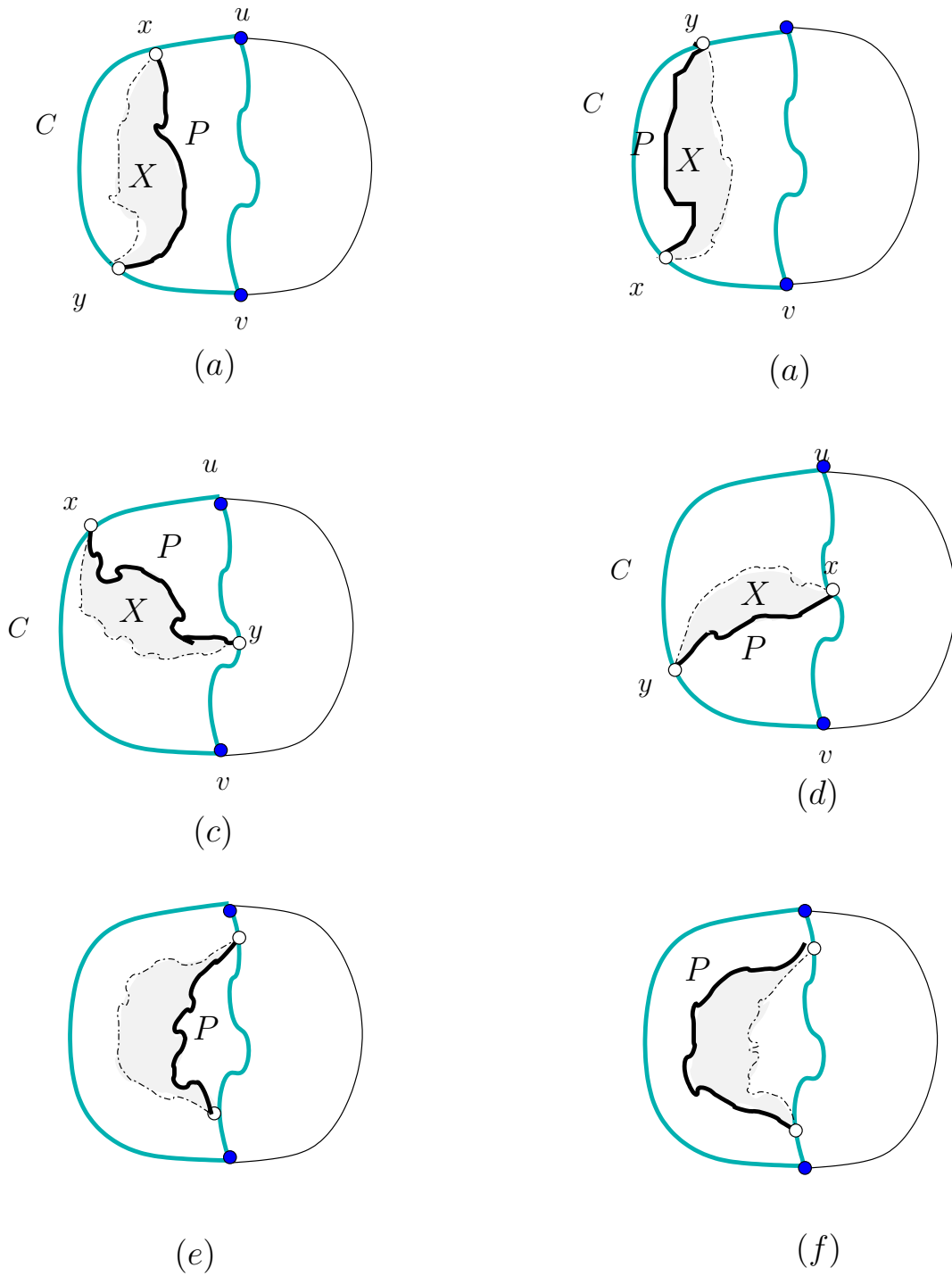


FIG. 3.6: Les configurations $A = \{(a), (b)\}$, $B = \{(c), (d)\}$ et $D = \{(e), (f)\}$.

Si le cas A est vérifié et qu'aucune des règles 1,2 et 3 ne peut s'appliquer, alors G contient un mineur $K_{2,4}$. En effet, par la règle 3, on a $|R| \geq 2$. Le chemin $L[x, y]$ n'est pas une arête car sinon les attachements de X seraient x, y et dans ce cas, la règle 2 aurait pu être appliquée. Donc, L est de longueur ≥ 2 . Par conséquent, nous

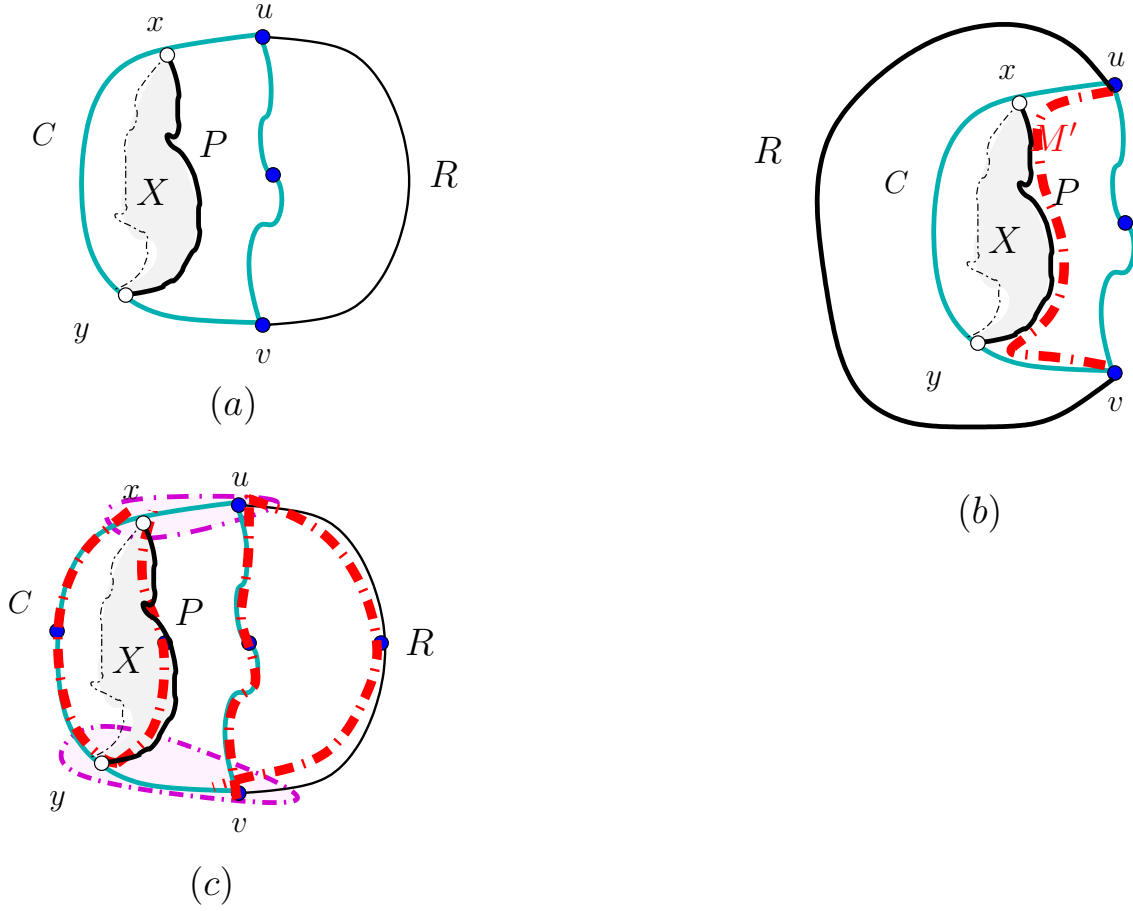


FIG. 3.7: Les configurations du cas A pour la règle 3.

avons 4 chemins disjoints de longueur ≥ 2 entre $A = L[x, u]$ et $B = L[y, v]$: les chemins $L[x, y]$, P , M et R (cf. figure 3.7 (c)). Ce qui termine le Cas A.

On suppose que le cas B correspond à $x \in L[u, v]$ et $y \in M[u, v]$ (cf. figure 3.6 (c)). Le cas $y \in L[u, v]$ et $x \in M[u, v]$ représenté par la figure 3.6 (d) peut se ramener au cas précédent en échangeant le rôle de u et v et le rôle de x et y .

Règle 4 : Si le cas B est vérifié et $M[u, y]$ est une arête, alors on étend R avec $L[x, u]$, c'est-à-dire on passe de (L, M, R) vers $(L[v, x], P \cup M[y, v], L[x, u] \cup R)$ et poser $u = x$. Ce nouveau plongement couvre au moins un sommet x' de plus ($x' \in X$) et donc ϕ croît strictement. Par exemple, la figure 3.8 (a) est le résultat de la règle 4 sur la figure 3.6 (c). Le chemin L est le chemin rouge, le chemin M est le chemin vert et le chemin R est le chemin bleu. On a la longueur du chemin R est ≥ 2 .

Règle 5 : Si le cas B est vérifié, $L[v, x]$ est une arête, et $\text{In}(M \cup R)$ ne contient aucune arête de $M[u, v]$ vers R , alors on bascule le chemin R vers la gauche, c'est-à-dire passer de (L, M, R) vers $(R \cup M[v, y], L[u, x] \cup P, M[u, x])$ et poser $v = y$. Nous pouvons facilement vérifier les quatre propriétés pour ce nouveau plongement.

Si $L[v, y]$ est une arête, la fonction de potentiel ϕ croît strictement car R est de longueur au moins deux (grâce à la règle 4), et les nouveaux chemins couvrent au moins un sommet x' de plus ($x' \in X$). Pour illustration, cf. figure 3.8 (b). Le chemin L du nouveau plongement est coloré en rouge, le chemin M en vert et le chemin R en bleu.

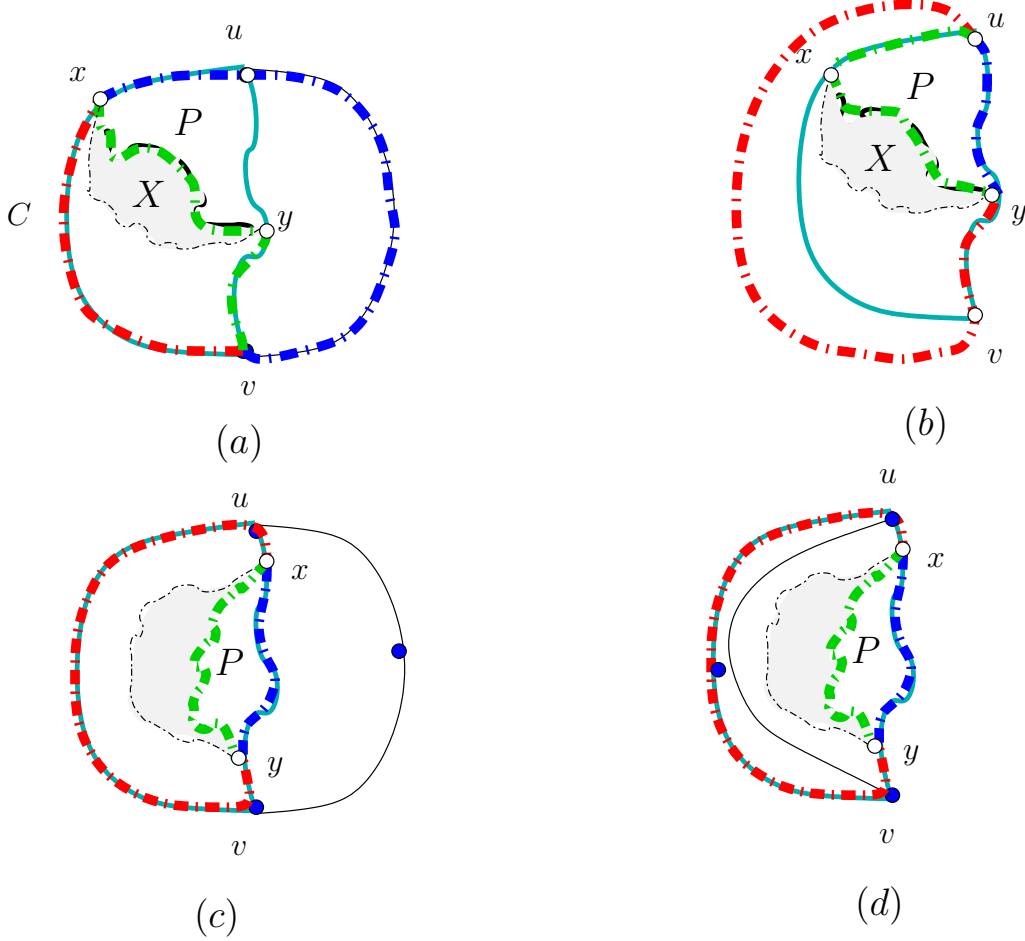


FIG. 3.8: Les configurations du cas B pour les règles 4 et 5.

Si le cas B est vérifié et qu'aucune des règles $1, \dots, 5$ n'est vérifiée, alors G contient un mineur $K_{2,4}$. En effet, grâce à la règle 4, $M[u, y]$ et par symétrie $M[v, y]$ sont de longueur ≥ 2 . Par la règle 5, soit 1) $L[v, x]$ et $L[x, u]$ sont tous les deux de longueur ≥ 2 ; soit 2) $\text{IN}(M \cup R)$ contient une arête wz telle que $w \in M]u, v[$ et $z \in R]u, v[$. Dans le cas 1) on construit le mineur $K_{2,4}$ avec $A = P$ et $B = R$. Les chemins $L[u, x]$, $M[u, y]$, $L[v, x]$, et $M[v, y]$ sont disjoints et sont de longueur ≥ 2 . Dans le cas 2) on construit le mineur $K_{2,4}$ avec $A = L[x, v]$ et $B = M[w, y]$. Les quatre chemins $L[x, w]$, $M[y, v]$, $\{wz\} \cup R[z, v]$, et $M[w, u] \cup L[u, x]$ sont disjoints et de longueur au moins deux. Ce qui termine le cas B .

Règle 6 : Si le cas C est vérifié, R ou L est une arête, $|M[x, y]| \geq 2$ et $\text{IN}(M \cup R)$ ne contient aucune arête de $M]u, v[$ vers R . Alors on bascule le chemin R vers la

gauche (comme dans la règle 3), c'est à dire passer de (L, M, R) vers $(M[y, v] \cup R \cup M[u, x], P, M[x, y])$ et poser $u = x$ et $v = y$ si R n'est pas l'arête. Si R est une arête, on bascule R vers la gauche dans la région $\text{IN}(L \cup M)$. Cela est possible tant qu'il n'existe pas une arête ou un chemin reliant L et M . L'existence d'un tel chemin ou d'une telle arête conduirait à l'existence d'un mineur $K_{2,4}$. En effet l'existence de cette arête conduirait à la configuration interdite (e) de la figure 3.9.

On vérifie que les quatre propriétés sont vérifiées. Notons que le nouveau chemin R (l'ancien chemin $M[x, y]$) est de longueur au moins deux. Si R était une arête, alors ϕ croit strictement car le nombre de sommets couverts augmente vu que maintenant $|R| \geq 2$. Si L est une arête, alors on a $|R| \geq 2$, car il n'y a pas d'arête multiple et dans ce cas, $|R|$ reste toujours au moins à deux. Cependant, les trois nouveaux chemins couvrent au moins un sommet de plus de X . Donc ϕ croit strictement dans ce cas (cf. figure 3.8 (c) et (d)).

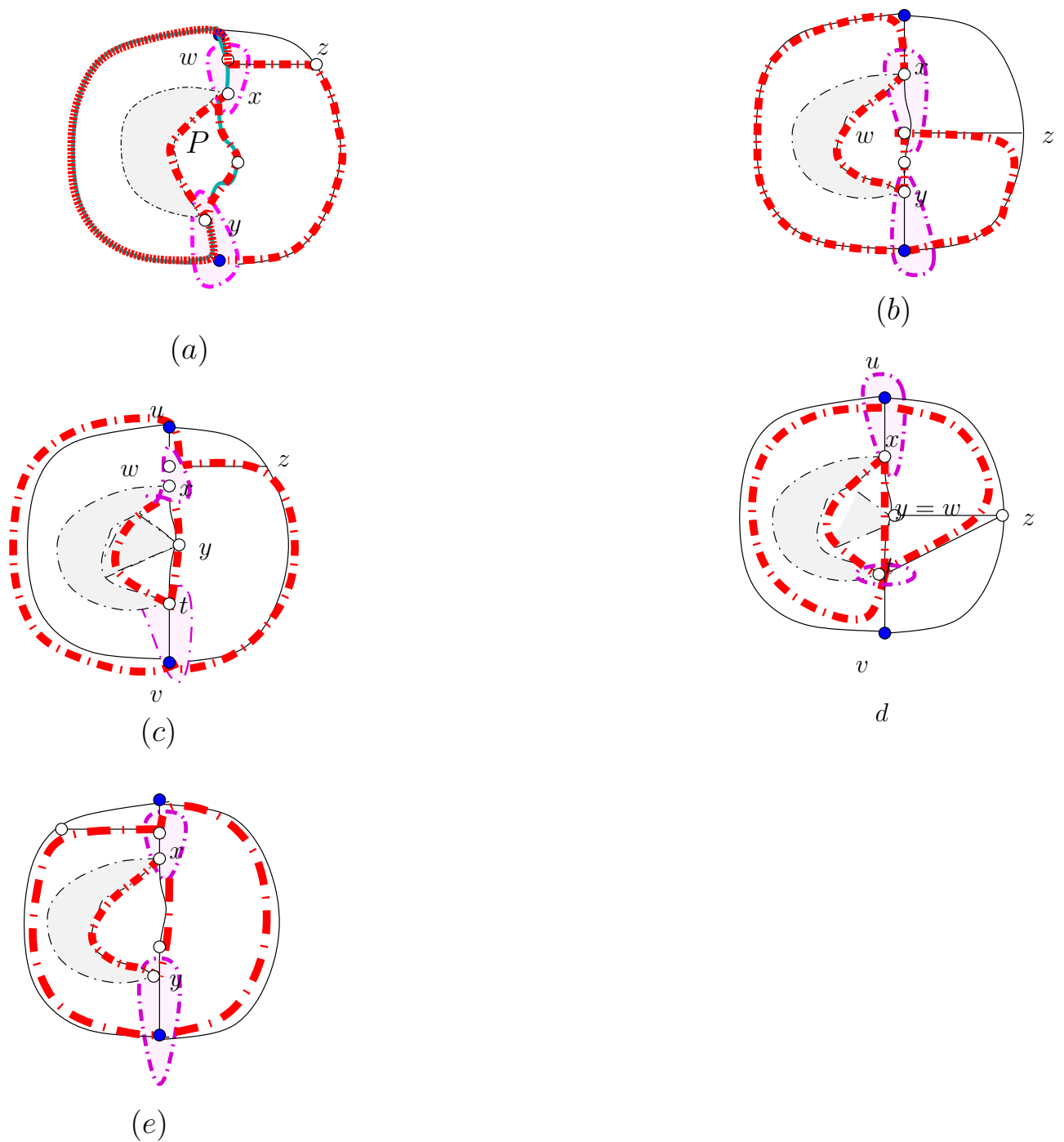
Si $|M[x, y]| = 1$ et $t \in A$ tel que $|M[y, t]| = 1$, alors on note par M' le chemin dans X entre x et t tel que $\text{IN}(M' \cup M[x, t])$ ne contient aucun sommet et aucune arête dont les deux extrémités sont dans M' . Le chemin M' existe car X est non vide, connexe et les sommets x, y et t sont des attachements disjoints. Donc, le chemin M' est de longueur au moins deux.

Alors, on bascule R vers la gauche et puis passer de (L, M, R) vers $(M[t, v] \cup R \cup M[u, x], M', M[x, t])$ (cf. figures 3.10 (a) et (b)). On vérifie facilement que le nouveau plongement LMR vérifie bien les quatre propriétés. En effet, $L \cup R$ représente la face extérieure. Le chemin M est de longueur au moins deux car constitué par le chemin M' . De plus, par la manière dont on a construit le chemin M' , on a $\text{IN}(M' \cup M[x, t])$ qui ne contient aucun sommet et aucune arête dont les deux extrémités sont dans M' .

On suppose qu'aucune des règles de 1 à 6 ne peut être appliquées. Nous pouvons souligner que dans ce cas le plongement de G possède beaucoup de contraintes dues au fait qu'aucune des règles précédentes ne peut être appliquées (cf. figure 3.9).

En particulier, le cas C est vérifié et $|R| \geq 2$ (par la règle 6). Nous avons aussi L qui est une arête car sinon, on construit un mineur $K_{2,4}$ avec $A = M[u, x]$ et $B = M[v, y]$ similairement avec la construction après la règle 3. Nous avons aussi $u \neq x$ ou $v \neq y$ car sinon on aurait $x, y \in L$ et donc le case A aurait dû être appliqué. On peut donc supposer sans perte de généralité que $u \neq x$. De plus, par le fait que L est une arête, la règle 6 implique que $\text{IN}(M \cup R)$ contient une arête wz entre un sommet $w \in M]u, v[$ vers un sommet $z \in R$, car sinon, on aurait du appliquer la règle 6. Si $w \in M]u, x[$ (ou strictement $w \in M]v, y[$) et que $|M[x, y]| \geq 2$, alors on construit un mineur $K_{2,4}$ avec $A = M[w, x]$ et $B = M[y, v]$. Les quatre chemins P , $M[x, y]$, $\{wz\} \cup R[z, v]$ et $\{wu\} \cup L[u, v]$ sont disjoints et de longueurs ≥ 2 (cf. figure 3.9 (a)). La construction du mineur est similaire si $w \in M]v, y[$. Par ailleurs, si $|M[x, y]| = 1$, alors par la règle 1, il existe un sommet $t \in A$ tel que $|M[x, t]| \geq 2$. Dans ce cas, on construit un mineur $K_{2,4}$ avec $A = M[x, w]$ et $B = M[t, v]$. Les quatre chemins $M[w, u] \cup L$, M' , $M[x, t]$ et $wz \cup R[z, v]$ sont disjoints et de longueur au moins deux (cf. figure 3.9 (c)).

Donc, nous pouvons supposer que l'arête wz de M vers R est telle que $w \in M]x, y[$. De même, $M[w, y]$ est une arête car sinon, on construit un mineur $K_{2,4}$ avec $A = M[x, w]$ et $B = M[y, v]$. Les quatre chemins P , $M[w, y]$, $\{wz\} \cup R[z, v]$,


 FIG. 3.9: Configurations interdites dans G .

et $M[x, u] \cup L$ sont disjoints et de longueurs au moins deux (cf. figure 3.9 (b)). On peut aussi supposer qu'il n'y a pas d'arête entre un sommet $t' \in M[y, v]$ et un sommet $z' \in R[z, v]$ car sinon, on construit un mineur $K_{2,4}$ avec $A = M[x, u]$ et $B = M[y, t'] \setminus y$. Les quatre chemins $M[t, v] \cup L$, M' , $M[x, t]$ et $\{t'z'\} \cup R[z, u]$ sont disjoints et de longueur au moins deux (cf. figure 3.9 (d)). On a l'une des

configurations (c) ou (e) de la figure 3.10.

Règle 7 : Si $M[x, y]$ n'est pas une arête, on construit un nouveau plongement-LMR et on met à jour (L, M, R) vers $(R[z, u] \cup M[u, x], P \cup M[y, v] \cup R[v, z], \{zw\} \cup M[w, x])$, et on pose $u = x$ et $v = z$ (cf. figure 3.10 (d)).

Sinon ($M[x, y]$ est une arête), on construit un nouveau plongement-LMR et on met à jour (L, M, R) vers $(R[z, u] \cup M[u, x], M' \cup M[t, v] \cup R[v, z], \{zw\} \cup M[w, x])$, et on pose $u = x$ et $v = z$ (cf. figure 3.9 (f)).

Il n'est pas difficile de voir que ϕ croît strictement puisque R est de longueur au moins deux et que les chemins L, M, R couvrent un sommet x' de plus (x' est un sommet de X) : les chemins qui étaient couverts avant et qui ne le sont plus dans le nouveau plongement sont L et $\{wz\}$, et sont des arêtes. De plus, nous pouvons facilement vérifier que le nouveau plongement satisfait bien les quatre propriétés. En effet, $L \cup R$ est bien le bord de la face extérieure.

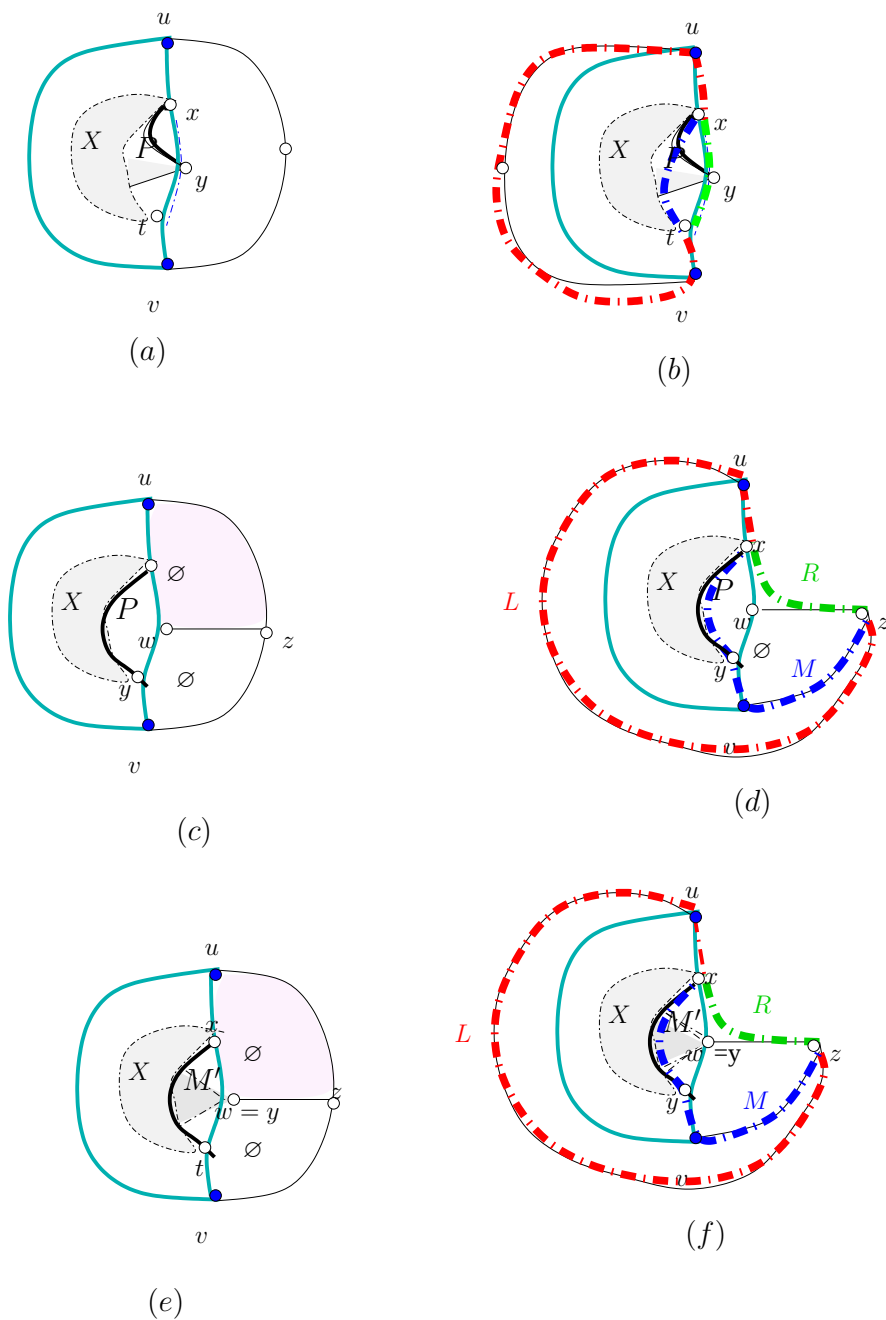


FIG. 3.10: Une configuration avec toutes les contraintes si aucune des règles 1 – 6 ne peut s'appliquer.

Sur la figure 3.9 (d), la face extérieure est constituée du chemin rouge L et du chemin vert (R). On a aussi $\text{IN}(M \cup R)$ qui ne contient aucun sommet. En effet, $\text{IN}(M \cup R)$ est engendrée par deux régions du plongement d'avant. La première région est comprise entre P et le chemin M du précédent plongement. Cette région est vide par définition de P ; elle ne contient ni d'arête ni de sommet. La deuxième région est engendrée par une partie de la région entre le chemin M et le chemin R dans l'ancien plongement. Par définition de M et R , cette région ne contient aucun

sommet. Donc $\text{IN}(M \cup R)$ ne contient aucun sommet. Pour ces mêmes raisons, cette région ne contient aucune arête dont les deux extrémités sont dans M . Par ailleurs, le nouveau chemin M contient P , alors que $|P| \geq 2$. Donc, le nouveau chemin M est de longueur au moins 2. Nous avons donc vérifié les quatre propriétés. Ceci termine la preuve du lemme 3.8. $\square$

3.3.3 Démonstration du lemme 3.5 : le cas non-planair

Pour prouver le lemme 3.5, nous avons besoin du lemme clef suivant :

Lemme 3.9 *Soit H une subdivision de $K_{3,3}$, un sous-graphe d'un graphe 2-connexe de G . Pour toute composante connexe X de $G \setminus H$, soit les attachements de X induisent une arête de H , soit on peut trouver en temps linéaire un mineur $K_{2,4}$ dans G .*

Démonstration : Le sous-graphe H est composé de deux ensembles de sommets de degré trois notés par I et J , et 9 chemins notés $P[i, j]$ reliant chaque sommet $i \in I$ à chaque sommet $j \in J$.

Soit X une composante connexe de $G \setminus H$, et notons A l'ensemble des attachements de X . On remarque que $|A| > 1$ car sinon le singleton A serait un séparateur de G : ce qui est impossible vu que G est 2-connexe.

Alors, on considère deux attachements $u, v \in A$. Le reste de la preuve consiste à montrer que si u et v sont à distance au moins deux dans H , alors on peut trouver en temps linéaire un mineur $K_{2,4}$ dans G . En d'autres termes, si G est sans mineur $K_{2,4}$, alors les attachements de X induisent exactement une arête de H .

Supposons que $u \in P[i_u, j_u]$ et $v \in P[i_v, j_v]$ avec $i_u, i_v \in I$ et $j_u, j_v \in J$, et que u et v sont distant qu'au moins deux dans H . Parmi les quatre longueurs $|P[i_u, u]|$, $|P[u, j_u]|$, $|P[i_v, v]|$ et $|P[v, j_v]|$, on supposera que $|P[i_u, u]|$ la plus petite. Cela est possible en échangeant soit u et v si le minimum est $|P[i_v, v]|$ ou $|P[v, j_v]|$ soit les i avec les j si $|P[i_u, u]| > |P[u, j_u]|$. Il y a en fait deux cas de figure :

Cas 1 : $i_u = i_v$ et $j_u = j_v$.

Les deux sommets u, v sont sur le chemin $P = P[i_u, j_v]$. Notons également qu'en partant de i_u vers j_v sur P on doit rencontrer u avant v à cause de la minimalité de $|P|$. Notons $A = P[i_u, u]$ et $B = P[v, j_v]$ les deux pôles du mineur $K_{2,4}$. Il existe deux chemins disjoints entre i_u et j_v dans H de longueur au moins deux ne contenant aucune arête de P . Le sous-chemin $P[u, v]$ est de longueur au moins deux vu que u et v sont distant au moins deux dans H . De plus, nous avons un quatrième chemin de longueur au moins deux entre u et v passant par la composante X , ce qui complète la construction du mineur $K_{2,4}$.

Cas 2 : $i_u \neq i_v$ ou $j_u \neq j_v$.

Nous présentons seulement la preuve pour le cas $i_u \neq i_v$; la preuve pour le cas $j_u \neq j_v$ peut être obtenue de manière similaire.

Si $v = j_v$, alors $u = i_u$ à cause de la minimalité de $|P[i_u, u]|$. En particulier, les deux sommets u, v sont sur le chemin $P[i_u, j_v]$ et donc nous pouvons conclure grâce

au cas 1. Par conséquent, nous pouvons supposer que le chemin $P[v, j_v]$ contient au moins une arête de H . Notons également que le sous-chemin $P[u, j_u]$ contient au moins une arête à cause de la minimalité de $|P[i_u, u]|$.

Posons $A = P[i_u, u]$ et $B = P[i_v, v]$ les deux pôles du mineur $K_{2,4}$. Comme discuté précédemment, les chemins $P[u, j_u]$ et $P[v, j_v]$ contiennent au moins une arête et par conséquent, la contraction de A et B dans H donne une subdivision de $K_{3,3}$, notée H' . Il existe trois chemins disjoints de longueur au moins deux entre A et B dans H' , et donc dans G . De plus, il existe un quatrième chemin de longueur au moins deux passant par X entre u et v . Ce qui complète donc la construction du mineur $K_{2,4}$.

Il est clair que dans chacun de deux cas de figure, le mineur $K_{2,4}$ peut être extrait en temps linéaire. $\square$

Nous allons maintenant donner la preuve du lemme 3.5.

Démonstration du lemme 3.5. Nous avons besoin de montrer qu'un graphe G 2-connexe et non planaire soit contient un mineur $K_{2,4}$ soit il contient un sommet v_0 tel que $G \setminus \{v_0\}$ est planaire. Pour cela, nous allons analyser une simple procédure appelées $\text{FIND}(G)$ qui soit retourne un mineur $K_{2,4}$ soit un sommet v_0 . Une implémentation de cette procédure en temps $O(n)$ sera donnée juste après cette preuve de correction.

Procédure $\text{FIND}(G)$:

1. Si $n \leq 5$, alors retourner un sommet quelconque de G comme étant v_0 .
2. Construire une subdivision H de K_5 ou $K_{3,3}$ dans G .
3. Si H est une subdivision de K_5 , alors retourner un mineur $K_{2,4}$ de H .
4. Tant que $H \neq G$:
 - 4a. Choisir une composante connexe X de $G \setminus H$.
 - 4b. Appliquer le lemme 3.9 à G et H . Si un mineur $K_{2,4}$ est trouvé, on le retourne. Sinon remplacer dans H l'arête $\{u, v\}$ induite par les attachements de X par un chemin de u à v passant par X .
5. Soit v_0 un sommet de degré 3 de H . Si $G \setminus \{v_0\}$ est planaire, retourner v_0 .
6. Construire une subdivision H' de K_5 ou $K_{3,3}$ dans $G \setminus \{v_0\}$.
7. Si H' est une subdivision de K_5 , alors retourner un mineur $K_{2,4}$ de H' .
8. Retourner un mineur $K_{2,4}$ construit lorsqu'on a appliqué le lemme 3.9 à G , H' et la composante connexe de $G \setminus H'$ contenant v_0 .

Correction Si on termine après l'étape 1, alors $G \setminus \{v_0\}$ a au plus 4 sommets, et donc il est planaire. Ce qui veut dire que l'étape 1 est correcte.

Comme G est non planaire, grâce au théorème de 1.3, G contient un sous graphe *homeomorphe* soit à K_5 ou $K_{3,3}$, c'est-à-dire une subdivision de K_5 ou de $K_{3,3}$. Un tel sous-graphe H peut être construit dans l'étape 2 en temps $O(n)$ [BM99, Wil84]. Si H est une subdivision de K_5 (étape 3), alors il existe dans H un chemin P de longueur au moins deux reliant deux sommets de degré 4 (du K_5). En effet, H contient plus de 5 sommets à cause de l'étape 1 et donc $H \neq K_5$. Alors on construit un mineur $K_{2,4}$ dans H de la manière suivante. Soit X l'ensemble des cinq sommets

de degré 4 de H (correspondant aux sommets du K_5), et soit $a, b \in X$ les deux extrémités du chemin P . Posons $A = \{a\}$ et $B = \{b\}$. Il existe trois chemins dans H entre a et b dont chacun utilise un sommet distinct de $X \setminus \{a, b\}$. Ces trois chemins sont de longueur au moins deux, deux-à-deux disjoints et ils sont aussi disjoints de P . Le chemin P est par construction un chemin de longueur au moins deux et est donc le quatrième chemin. Ce qui complète la construction du mineur $K_{2,4}$ et prouve donc la correction de l'étape 3. Il est clair que ces deux constructions peuvent être réalisée en temps linéaire en la taille de H qui est $O(n)$.

Nous allons finir avec le cas où H est une subdivision de $K_{3,3}$ dans l'étape 4. Nous observons que si l'étape 4b ne sort pas avec un mineur $K_{2,4}$, alors le nombre de sommets de H augmente de un car une arête de H est remplacée par un chemin de longueur au moins deux passant par X . Ce qui veut dire que soit on sort par l'étape 4b en retournant un mineur $K_{2,4}$, soit on sort par l'étape 5 avec un sous-graphe H couvrant G qui est une subdivision de $K_{3,3}$. On remarque qu'à cette étape, G est 2-connexe car H l'est.

Soit $G' = G \setminus \{v_0\}$, avec v_0 un sommet de degré 3 de H . Supposons que G' n'est pas planaire, ce qui veut dire que la procédure ne termine pas par l'étape 5. Les étapes 6 et 7 sont correctes vu que ce sont les mêmes que les étapes 2 et 3 en remplaçant G, H par G', H' . Dans l'étape 8 on applique le lemme 3.9 à G et H' car G est 2-connexe. Nous affirmons que l'application de ce lemme retourne un mineur $K_{2,4}$ de G .

Considérons une composante connexe X_0 de $G \setminus H'$ contenant v_0 . Si nous ne retournons un mineur $K_{2,4}$, alors c'est parce que les attachements u, v de X_0 induisent une arête de H' . En particulier $\{u, v\}$ déconnecte G en $X_0 \ni v_0$ et une autre composante nommée Y contenant les autres sommets de H' (il y en a au plus cinq sommets). Il s'en suit que $\{u, v\}$ est un v_0 -séparateur de G . Cependant, dans H , il existe un chemin de v_0 vers Y . Alors, $\{u, v\}$ n'est pas un v_0 -séparateur dans H et donc dans G : ce qui est une contradiction. Ainsi, l'étape 8 est correcte et complète la preuve de correction de la procédure FIND(G).

Une implémentation en temps $O(n)$. La procédure FIND(G) ne peut pas être utilisée directement si on veut une complexité en temps de $O(n)$ et cela pour au moins deux raisons. Premièrement, le graphe G donné en entrée peut avoir $m = \Omega(n^2)$ arêtes. Deuxièmement, la boucle "tant que" dans l'étape 4 peut faire $\Theta(n)$ boucles, dont chacune s'effectue en temps $\Theta(m)$.

En utilisant le résultat de [CRS08], nous savons que si G contient plus de $(n - 1)(r + 1)/2 = 2.5n - O(1)$ arêtes, alors G contient un mineur $K_{2,4}$. Cependant, cela n'implique pas que la procédure FIND(G) retourne un tel mineur. Cela est du au fait que notre procédure peut trouver un sommet v (de grand degré) tel que $G \setminus \{v\}$ est planaire même si G contient plus de $2.5n$ arêtes.

Cependant, si G a au moins $4n - 6$, alors l'application de FIND(G) retourne nécessairement un mineur $K_{2,4}$. Cela parce que $G \setminus \{v\}$, pour tout sommet v ne peut pas être planaire : $G \setminus \{v\}$ peut contenir au moins $4n - 6 - (n - 1) = 3n - 5$ arêtes. En d'autres termes, si G contient beaucoup d'arêtes, on peut concentrer notre attention dans le sous graphe de G ayant $4n - 6$ arêtes et on applique la dessus une rapide

implémentation de FIND. L'extraction de ce sous-graphe peut être faite en temps $O(n)$ en sélectionnant ses $4n - 6$ premières arêtes. Ainsi, nous pouvons supposer que G a $O(n)$ arêtes.

Dans la preuve de la correction de FIND, nous avons vu que chaque étape excepté l'étape 4 s'effectue en temps linéaire, et donc $O(n)$. L'étape 4b s'effectue en $O(n)$ si un mineur $K_{2,4}$ est construit. Sinon, par la 2-connectivité de G , un simple DFS partant des sommets de X trouvera un chemin de u vers v en temps proportionnelle à la longueur du chemin. En d'autres termes, chaque arête est visitée $O(1)$ fois, et donc la boucle "tant que" fait au total $O(n)$. Ce qui complète la preuve du lemme 3.5. $\square$

3.3.4 Généralisation du théorème 3.2

Pour tout entier $k \geq 0$, un graphe k -apex planaire-extérieur est un graphe possédant k sommets dont la suppression rend le graphe planaire-extérieur. Ainsi, les graphes 0-apex planaire-extérieurs sont les graphes planaire-extérieurs, les graphes 1-apex planaire-extérieurs sont des graphes planaires, et les graphes K_5 et $K_{3,3}$ sont des graphes 2-apex planaire-extérieurs, mais ils ne sont pas des graphes 1-apex planaires. Pour toute famille $\mathcal{F}$ de graphes, un *arbre de $\mathcal{F}$* , est un graphe dont les composantes 2-connexes sont dans $\mathcal{F}$.

Le théorème 3.2 prouve que tout graphe sans $K_{2,4}$ comme mineur est un arbre de graphes 2-apex planaire-extérieur. Nous avons la généralisation suivante :

Corollaire 3.10 *Tout graphe sans mineur $K_{2,k+2}$ est un graphe k -apex planaire-extérieur pour tout entier $k \in \{0, 1, 2\}$.*

Démonstration : Pour $k = 0$, il est évident de voir que tout graphe excluant $K_{2,2}$ comme mineur est planaire-extérieur car ce graphe est un arbre de K_3 , et donc c'est un graphe 0-apex planaire-extérieur.

Un graphe qui n'est pas planaire-extérieur contient forcément une subdivision de soit $K_{2,3}$ soit K_4 comme sous-graphe. Il est facile de voir que toute subdivision de K_4 contient $K_{2,3}$ comme mineur. Par conséquent, toute composante 2-connexe d'un graphe excluant $K_{2,3}$ comme mineur est soit K_4 ou un graphe planaire-extérieur. On remarque que K_4 est un graphe 1-apex planaire-extérieur, ce qui à pour conséquence que tout graphe sans mineur $K_{2,4}$ est un arbre de graphes 1-apex planaire-extérieurs, ce qui fait donc que le corollaire est vrai pour $k = 1$.

Pour $k = 2$, nous pouvons conclure grâce au théorème 3.2. $\square$

Malheureusement, le corollaire 3.10 n'est pas vérifié pour $k = 3$, c'est-à-dire pour les graphes sans mineur $K_{2,5}$.

Théorème 3.11 *Pour tout entier $k \geq 0$, il existe un graphe sans mineur $K_{2,5}$ et ce graphe n'est pas un arbre de graphes k -apex planaire-extérieurs. Cependant ce graphe est planaire et a $4k + 4$ sommets.*

Démonstration : Pour tout entier $p \geq 1$, on définit le graphe G_p comme suit :

1. G_1 est un cycle de longueur 4 plus une arête connectant deux sommets non adjacents du cycle.
2. Pour tout entier $p \geq 2$, le graphe G_p est obtenu à partir de G_{p-1} en ajoutant deux sommets connectés à chaque sommet de degré deux du graphe G_{p-1} .

Il est facile de montrer par induction que le graphe G_p a exactement deux sommets de degré deux, il est 2-connexe, il possède $2p + 2$ sommets et possède un plongement planaire sur le plan comme présenté sur la figure 3.11.

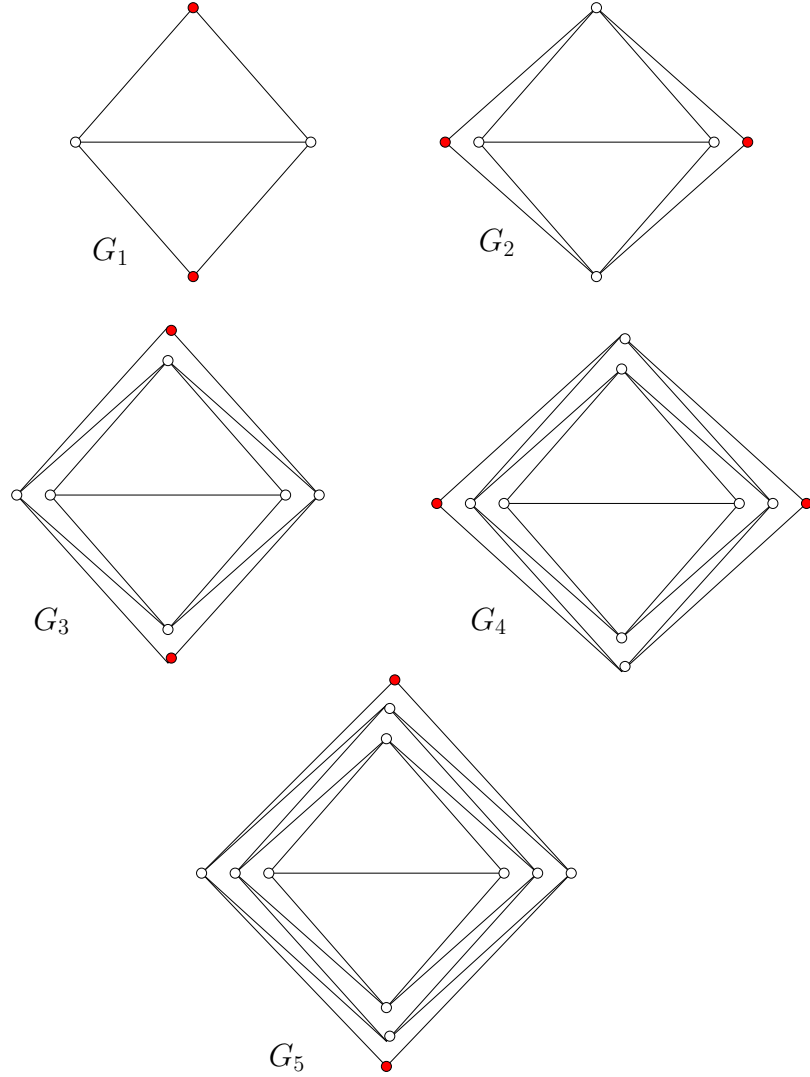


FIG. 3.11: Dessin planaire de $G_1 = H_0$, G_2 , $G_3 = H_1$, G_4 , et $G_5 = H_2$. Les sommets de degré deux sont colorés en rouge.

Pour tout entier $k \geq 0$, on pose $H_k = G_{2k+1}$. Nous allons montrer que le graphe H_{k+1} ayant $4k + 4$ sommets n'est pas un graphe k -apex planaire-extérieur.

On remarque que si on supprime le cycle C_k de longueur 4, bordant la face extérieure de H_k , on obtient le graphe H_{k-1} si $k > 0$. Donc les sommets de H_k sont

partitionnés en $k+1$ couches de cycles $C_k, C_{k-1}, \dots, C_0$, telles que la dernière couche de cycle C_0 correspond au graphe $H_0 = G_1$.

Montrons que le graphe H_k est sans mineur $K_{2,5}$. Supposons que H_k contient $K_{2,5}$ comme mineur. Notons A et B les sous-graphes connexes de H_k correspondant aux deux sommets de degré 5 du mineur $K_{2,5}$. Chacun des sous-graphes A et B intersecte des couches de cycles C_i . Notons $[i_A, j_A]$ le segment des indices des couches de cycles intersectant A et $[i_B, j_B]$ le segment des indices des couches de cycles intersectant B . Nous pouvons supposer sans perdre de généralité que $i_A \leq i_B$.

Cas $j_A < i_B$: il y a quatre chemins disjoints connectant C_{j_A} à C_{j_A+1} et donc C_{i_B} . D'où, il n'y a pas cinq chemins disjoints entre A et B : ce qui est une contradiction.

Cas $i_B \leq j_A$: soient A' et B' les parties de respectivement A et B induite par les cycles dont les indices forment le segment $[i_A, j_A] \cap [i_B, j_B]$. Quel que soit la taille de ce segment, il y a au plus deux chemins disjoints entre A' et B' (à travers un seul cycle C_t de ce segment). D'après le cas précédent, nous avons vu que : que ce soit entre $A \setminus A'$ et B , ou similairement entre $B \setminus B'$ et A , il y a au plus quatre chemins disjoints. Cependant, ce second ensemble de chemins intersecte C_t . Seul deux chemins peuvent relier à un sommet de C_i avec $i < t$, et deux chemins peuvent connecter à un sommet de C_j avec $j \geq t$. Ce qui fait donc un total d'au plus quatre chemins disjoints entre A et B : ce qui est une contradiction.

Le graphe H_k est sans mineur $K_{2,5}$.

Montrons que le graphe H_k n'est pas k -apex planaire-extérieur. Soit X un ensemble de sommets tels que $H_k \setminus X$ est planaire-extérieur. Nous allons montrer que $|X| \geq k$. Nous dirons qu'un cycle C_i est *sûr* si $C_i \cap X = \emptyset$. Sinon, on dira que le cycle est *non-sûr*. Une *zone interne* est un segment $[a, b]$ constitué d'au moins trois indice de couches de cycles telle que C_a et C_b sont les seules cycles sûres parmi tous le cycles dont les indices sont dans ce segment.

Soit s le nombre de cycles sûres, et z le nombre de zones internes définies par X dans H_k . Il est évident que $s + z \leq k + 1$ vu qu'il y a au moins un cycle non-sûr par zone interne. Cependant, le nombre de zones internes peut être plus petit que le nombre de cycles sûres, c'est-à-dire $z < s$.

Considérons une zone interne $[a, b]$. Cette zone contient $b - a - 1 \geq 1$ cycles non-sûres. Au moins un de ces cycles non-sûres contient deux sommets de X ou plus. Cela est du au fait qu'entre deux sommets de degré > 2 du cycle C_b , il y a quatre chemins disjoints de longueur deux dans le sous graphe $H_k[C_b \cup C_{b-1}]$. Ainsi, nous avons besoin de supprimer au moins deux sommets du cycle C_{b-1} pour supprimer le $K_{2,3}$. D'où, le segment $[a, b]$ contribue pour au moins $(b - a - 1) + 1$ sommets de X , c'est-à-dire un de plus que le nombre de cycles non-sûres qu'il contient. En général, $|X| \geq (k + 1 - s) + z$, le terme $k + 1 - s$ devient le nombre total de cycles non-sûres.

Une observation importante est que le graphe induit par deux cycles sûres consécutifs, notés C_i, C_{i+1} , contient un mineur $K_{2,4}$. Donc, si $s > z + 1$, alors il existe deux cycles sûres consécutifs, et $H_k \setminus X$ contient un $K_{2,3}$ ($K_{2,4}$ dans notre cas), et il n'est pas planaire-extérieur : ce qui est une contradiction. Donc on a $s \leq z + 1$. Nous avons vu que $z < s$, par conséquent $z = s - 1$.

Il s'en suit que $|X| \geq k + 1 - s + z$ avec $z = s - 1$, par conséquent $|X| \geq k$. En d'autres termes, pour tout entier $k \geq 0$, le graphe H_{k+1} n'est pas k -apex planaire-extérieur. $\square$

3.4 La borne supérieure de la largeur arborescente des graphes planaires

Dans cette section nous nous intéressons à la structure des graphes planaires qui excluent un mineur H dessinable sur une grille $p \times q$. Un graphe H est dessinable sur une grille $p \times q$ s'il admet un dessin sur cette grille tel que les arêtes de H sont représentées par des segments de droite et les sommets de H sont représentés par des points de la grille. La taille de la grille est déterminée par le plus petit rectangle qui couvre le dessin de H . La longueur p de la grille est la longueur du rectangle et la largeur q de la grille est la largeur du rectangle. La taille de la grille est habituellement décrite par $p \times q$ et la grille est notée $G_{p \times q}$. Le problème de dessiner un graphe planaire sur une grille de taille minimale est un vrai challenge. Plusieurs travaux existent dans ce domaine [CiN95, DPP90, Sch90]; par exemple, tout graphe planaire à n sommets est dessinable sur une grille $p \times q$ avec $p \times q \leq (n-1) \times (n-1)$.

Plus précisément, nous consacrerons cette partie à la démonstration du théorème 3.3. Ce résultat améliore la borne actuelle connue pour la largeur arborescente des graphes planaires excluant un graphe planaire dessinable sur une grille $p \times q$ comme mineur qui est $O(r)$ [RST94]. Le résultat de Robertson *et al.* est une combinaison des deux lemmes suivant :

Lemme 3.12 [RST94]

Tout graphe planaire G excluant une grille $s \times s$ (avec s un entier ≥ 1) comme mineur est de largeur arborescente au plus $6s - 5$.

Lemme 3.13 [RST94] *Tout graphe planaire H peut être obtenu comme mineur d'une grille $r \times r$ avec $r = 14|V(H)| - 24$.*

Nous montrons le lemme clef suivant :

Lemme 3.14 *Tout graphe planaire H dessinable sur une grille $p \times q$ peut être obtenu comme mineur d'une grille $k \times k$ avec $k \leq 4p^{3/2}\sqrt{q} + p - 1$.*

3.4.1 Démonstration du théorème 3.3

En combinant le lemme 3.14 et le lemme 3.12, nous montrons que si G est sans mineur H , alors il existe un entier $k \leq 4p^{3/2}\sqrt{q} + p - 1$ tel que $\text{tw}(G) \leq 6k \leq 24p^{3/2}\sqrt{q} + 6(p - 1)$.

Donc, on a montré que $\text{tw}(G) = O(p^{3/2}\sqrt{q})$. Par conséquent, on a montré le théorème 3.3.

Par conséquent, les graphes sans mineur $K_{2,r}$ sont de largeurs arborescentes au plus $O(\sqrt{r})$. En effet, comme le montre la figure 3.12, tout graphe $K_{2,r}$ est dessinable sur une grille $3 \times r$. Ce résultat améliore la borne actuelle qui est de $r + 2$ [Thi99]. De

plus, ce résultat est asymptotiquement optimal car la grille $k \times k$ exclut clairement K_{2,k^2} comme mineur, et elle est de largeur arborescente k .

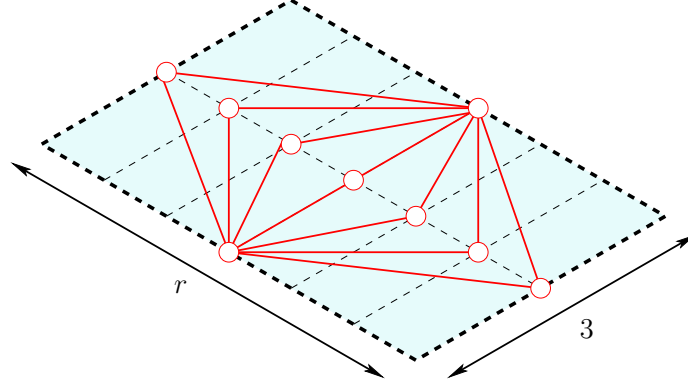


FIG. 3.12: Dessin d'un $K_{2,r}$ sur une grille $3 \times r$.

3.4.1.1 Démonstration du lemme 3.14

Soit H un graphe dessinable sur une grille $p \times q$. Pour montrer le lemme 3.14, nous allons construire étape par étape trois graphes H' , H'' et H''' tels que $H < H' < H'' < H'''$ et chacun de ces graphes possède une représentation particulière sur une grille éventuellement plus grande que la grille $p \times q$. Ensuite, dans la dernière étape, nous allons montrer qu'il existe un entier $k \leq 4p^{3/2}\sqrt{q} + p - 1$ et une grille $k \times k$ contenant H''' comme mineur. Vu que $H < H'''$, alors la grille $k \times k$ contient H comme mineur.

On suppose que les lignes de la grille sont numérotées de 1 à p et les colonnes de 1 à q . Ainsi, chaque point de la grille est identifié par un couple d'indices (i, j) représentant respectivement l'indice de la ligne et l'indice de la colonne. Chaque sommet $x \in H$ dont l'emplacement sur la grille est le point (i, j) sera noté $x_{i,j}$. Le point $(1, 1)$ est le coin supérieur gauche de la grille et le point (p, q) , et le coin inférieur droit de la grille.

Pour chaque sommet $x_{i,j} \in H$ appartenant à une ligne i dans la grille $p \times q'$, on note par $V_N(x_{i,j})$ et $V_S(x_{i,j})$ l'ensemble des voisins de $x_{i,j}$ sur les lignes respectivement inférieures à i et supérieures à i . De même, on note par $V_E(x_{i,j})$ et $V_O(x_{i,j})$ l'ensemble des voisins de $x_{i,j}$ sur les colonnes respectivement inférieures à j (à l'ouest) et supérieures à j (à l'est).

On peut remarquer que pour tout sommet $x_{i,j}$ de H , on a $|V_E(x_{i,j})| \leq 1$ et $|V_O(x_{i,j})| \leq 1$, car sinon on aurait des arêtes qui ne sont pas de segments de droite.

Étape 1 : Construction de H' à partir de H Cette étape consistera à montrer le lemme suivant :

Lemme 3.15 *Il existe un graphe H' contenant H comme mineur, et H' est dessinable sur une grille $p \times q'$ de telle sorte que pour tout sommets $x_{i,j} \in H'$ et $y_{k,l} \in V_N(x_{i,j}) \cup V_S(x_{i,j})$, on a $|i - k| = 1$.*

Démonstration : En effet, soit H' un graphe construit à partir de H en remplaçant dans H' toute arête $e_{i,j} = (u_{i,k}, v_{j,l}) \in H$ par un chemin de longueur $|i - j|$. Par exemple, sur la figure 3.13, en (a) on a un dessin d'un graphe H et en (b) on a un dessin du graphe H' construit à partir de H en utilisant cette méthode. Les sommets de $H' \setminus H$ sont les sommets blancs et les sommets noirs sont les sommets de $H' \cap H$. Sur cet exemple, la grille sur laquelle est dessiné H est la même que celle où on a dessiné H' . On remarque que chaque arête $e_{i,j} \in H$ induit dans H' au plus $|i - j|$ sommets supplémentaires. Par exemple, l'arête entre le sommet $x_{1,3}$ et le sommet $x_{5,1}$ sur la figure 3.13 (a), est remplacée par un chemin de longueur 4 sur la figure 3.13 (b).

Étant donné que H est planaire, alors le nombre d'arêtes de H est au plus $3|H| - 6$ d'après la formule d'Euler. Or, on a p lignes dans la grille $K_{p \times q}$. Donc, dans le pire des cas, chaque arête de H relie un sommet de la ligne 1 à un sommet de la ligne p et donc induit un chemin p_0 de longueur au plus $p - 1$ dans H' . Vu que les deux extrémités de p_0 sont des sommets de H , alors p_0 induit au plus $p - 1$ sommets de plus dans H' . Par conséquent, le nombre maximum de sommets créés dans H' est $\leq (p - 1)(3|H| - 6) = 3p|H| - 6p - 3|H| + 6$. Or on a $V(H) \subseteq V(H')$. D'où, on a $|H'| \leq 3p|H| + 6 - (6p + 2|H|) \leq 3p|H| + 6$. Vu que la seule opération effectuée est la subdivision des arêtes de H , alors H est un mineur de H' . Pour représenter H' sur une grille, on peut considérer la représentation de H sur la grille $K_{p \times q}$ et ensuite représenter les nouveaux sommets créés sur cette grille en augmentant la taille de la grille s'il le faut. En effet, les sommets de chaque chemin p_e de H' induit par une arête e de H peuvent être représentés sur une colonne de la grille. Ainsi, le nombre de ligne de la grille reste inchangé. Si toutes les colonnes de $K_{p \times q}$ sont déjà occupées, alors on crée de nouvelle colonne jusqu'à ce que tous les sommets de H' soient tous représentés. Dans le pire des cas, pour chaque chemin, il faut créer une nouvelle colonne. Donc, le nombre de colonnes nouvellement créées pour représenter H' dans une grille est au plus $3|H| - 6$. D'où, $q' \leq q + (3|H| - 6)$. Or $|H| \leq pq$. Ce qui implique que $q' \leq q + 3(pq - 2) = q + 3pq - 6p \leq 4pq$. Dans cette grille, chaque sommet $x_{i,j}$ de H' est adjacent qu'à des sommets sur les lignes $i - 1, i, i + 1$. Nous avons par conséquent montré le lemme 3.15. $\square$

Étape 2 : Construction de H'' à partir de H' Cette étape sera essentiellement consacrée à la preuve du lemme suivant :

Lemme 3.16 *On peut construire un graphe H'' à partir de H' de telle sorte que H'' soit dessinable sur une grille $p \times q''$ et que pour tout sommet $x \in H''$, on a $|V_E(x)| \leq 1$, $|V_O(x)| \leq 1$, $|V_N(x)| \leq 1$ et $|V_S(x)| \leq 1$ et $q'' \leq 3q'$.*

Démonstration : Soit H'' le graphe obtenu à partir de H' en remplaçant chaque sommet $x_{i,j} \in H'$ par un chemin $p_{x_{i,j}}$ de longueur $\max\{|V_N(x_{i,j})|, |V_S(x_{i,j})|\}$ de la manière suivante :

d'abord le chemin $p_{x_{i,j}}$ est représenté sur la ligne i , une extrémité v_1 de $p_{x_{i,j}}$ est reliée par une arête à $V_E(x_{i,j})$ et l'autre extrémité v_2 de $p_{x_{i,j}}$ est reliée par une arête à $V_O(x_{i,j})$. Par hypothèse, on $|V_E(x_{i,j})| \leq 1$ et $|V_O(x_{i,j})| \leq 1$.

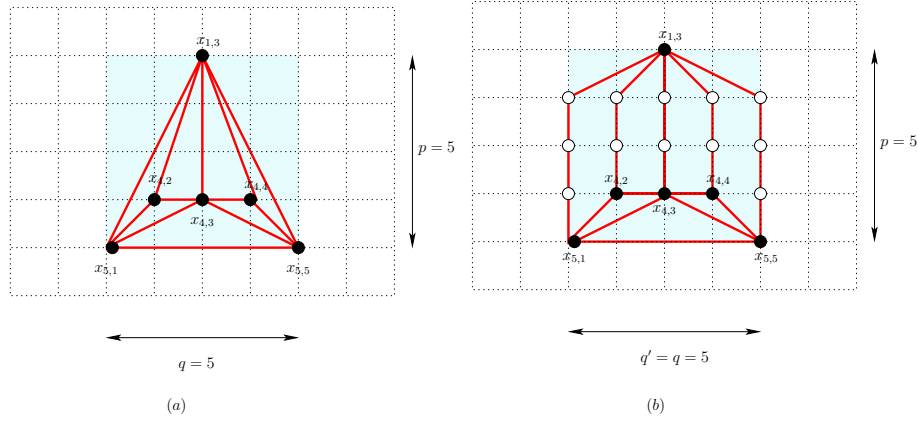


FIG. 3.13: (a) Un graphe H dessiné sur une grille $K_{4 \times 4}$. (b) Le dessin du graphe H' construit à partir de H par le lemme 3.15.

en suite, on relie de manière planaire tout sommet de $V_N(x_{i,j}) \cup V_S(x_{i,j})$ à un sommet de $p_{x_{i,j}}$ de telle sorte que pour tout sommet $y \in p_{x_{i,j}}$, on ait $|V_N(y)| \leq 1$ et $|V_S(y)| \leq 1$. Cette opération est bien possible du fait de la planarité de H' et donc de $\{x_{i,j}\} \cup N(x_{i,j})$. Par exemple sur la figure 3.14, en (b) on a le graphe H'' obtenu à partir du graphe H' représenté en (a) grâce à cette méthode. On remarque que le sommet bleu de la figure (a) est remplacé par un chemin avec 3 sommets dans la figure (b) dont les 3 sommets sont colorés en bleu. Nous remarquons que les nouveaux sommets introduits pour former le chemin $p_{x_{i,j}}$ occupe la même ligne que celle de $x_{i,j}$. Donc, le nombre de lignes reste inchangé dans H' et dans H'' . Par contre, le chemin $p_{x_{i,j}}$ ainsi formé augmente le nombre de sommets sur la ligne de $x_{i,j}$. Donc, le nombre de colonnes augmente de $|p_{x_{i,j}}|$ colonnes. En appliquant cette opération sur tous les sommets de H' , on construit un dessin du graphe H'' tel que pour tout sommet $x \in H''$, on a $|V_E(x)| \leq 1$, $|V_O(x)| \leq 1$, $|V_N(x)| \leq 1$ et $|V_S(x)| \leq 1$ et $q'' \leq 3q'$. De plus, le nombre de lignes de la grille est p .

Calculons le nombre q'' de colonnes de la grille. On remarque que tout sommet $x_{i,j} \in H'$ induit dans H'' un chemin $p_{x_{i,j}}$ de longueur $\max\{|V_N(x_{i,j})|, |V_S(x_{i,j})|\}$. La longueur de la ligne i augmente donc de $|p_{x_{i,j}}|$ colonnes. Ainsi si cette ligne avait q' sommets, alors elle contiendrait $q' + |p_{x_{i,j}}|$ sommets dans H'' . Vu qu'une opération sur un sommet $x_{i,j}$ n'agit que sur la longueur de la ligne L_i contenant $x_{i,j}$, alors un ensemble d'opérations sur un ensemble quelconque $\{x_1, x_2, \dots, x_k\}$ de sommets appartenant à des lignes différentes produirait dans H'' un ensemble $\{P_1, P_2, \dots, P_k\}$ de chemins et le nombre de colonnes de H'' serait incrémenté d'au plus $\max\{|P_1|, |P_2|, \dots, |P_k|\}$.

Le pire des cas qu'on pourrait imaginer est que tous les sommets soient sur une même ligne L_i . Soit L_{i-1} et L_{i+1} respectivement la ligne $i - 1$ et $i + 1$. Alors $|N(L_i)| \leq |L_{i-1}| + |L_{i+1}| \leq 2q'$. Or, cette opération remplace chaque sommet $x_{i,j}$ par un chemin de longueur au plus $|N(x_{i,j})|$. Donc, la somme des longueurs de tous les chemins produits sur L_i ne peut pas excéder $2q'$. Par conséquent, nous avons augmenté le nombre de colonne d'au plus $2q'$. Donc, on a $q'' \leq 3q'$. Ce qui termine la preuve du lemme 3.16. $\square$

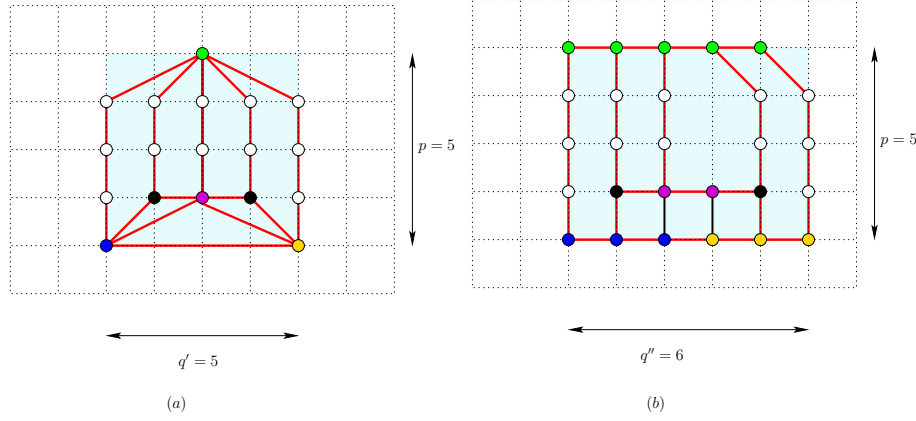


FIG. 3.14: (a) Le dessin du graphe H' de la figure 3.14 (a) sur lequel les sommets de niveau i dans H qui possèdent plus de deux voisins dans les niveaux $i - 1$ et $i + 1$ sont représentés chacun par une couleur. (b) Un dessin du graphe H'' construit à partir de H' par le lemme 3.16. Les sommets appartenant au chemin $p_{x_{i,j}}$ induit par le sommet $x_{i,j}$ sont colorés avec la même couleur que $x_{i,j}$. Par exemple le sommet coloré en vert sur la figure (a) induit un chemin de longueur 4 dont tous les sommets de ce chemin sont colorés en vert sur la figure (b)

En combinant le lemme 3.16 et le lemme 3.15, on a $q'' \leq 12pq$. De plus, le graphe H'' ainsi obtenu est de degré ≤ 4 .

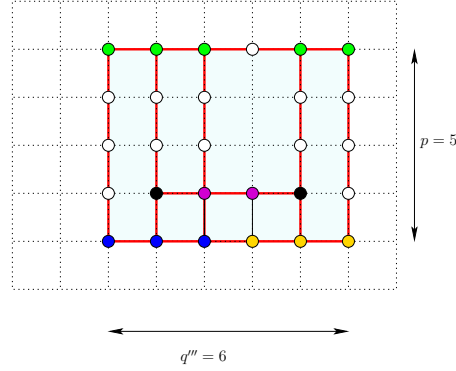
Étape 3 : Construction de H''' à partir de H'' Dans cette étape, nous montrons le lemme suivant :

Lemme 3.17 *A partir du dessin de H'' , on peut construire un dessin d'un graphe H''' sur une grille $p \times q'''$ tel que $H'' < H'''$ et chaque arête de H''' correspond à un segment de ligne droite de la grille. De plus $q''' \leq pq''$.*

Démonstration : Par définition du dessin de H'' , toute arête reliant deux sommets de la même ligne est un segment de ligne droite de longueur 1. Donc, toute arête qui n'est pas un segment de droite de la grille est une arête reliant deux sommets appartenant à deux lignes différentes de la grille. Ces segments sont donc obliques.

La construction de H''' consiste comme sur la figure 3.15 à redresser tout segment oblique en un segment vertical en décalant une des extrémités du segment. En effet, considérons un segment oblique $e = (u_{i,j}, v_{k,l})$. Par définition du dessin de H'' , on a soit $k = i + 1$ soit $k = i - 1$. On peut supposer sans perdre de généralité que $k = i + 1$. De même par le fait que e est oblique, on a soit $j < l$ soit $l < j$. On peut supposer sans perdre de généralité que $j < l$. Le décalage va consister à transférer le sommet $u_{i,j}$ en position (i, l) de la grille et à faire un décalage pour tout les sommets qui étaient sur une colonne $m \geq k$. Ce décalage augmente donc la longueur de la ligne L_i de $l - j$. Cette valeur est maximal si $l = 1$ et $j = q''$. En d'autres termes, le décalage maximal effectué entre deux couples de lignes est q'' . Ainsi, vu que le nombre de lignes est p , alors une ligne peut effectuer des décalage avec au plus $p - 1$

lignes. Donc, la longueur maximale d'un décalage est $\leq (p-1)q''$. Donc, la ligne la plus longue après avoir effectué ces décalages est de longueur $\leq (p-1)q'' + q'' = pq''$. Or, $q'' \leq 12pq$. D'où, on a $q''' \leq 12p^2q$.



□

FIG. 3.15: Un dessin du graphe H''' construit à partir de H'' .

Étape 4 : Construction d'une grille $k \times k$ contenant H''' On a besoin du lemme suivant pour construire la grille $k \times k$:

Lemme 3.18 *Toute grille $p \times q$ est mineur d'une grille $k \times \lceil \frac{q-2(p-1)}{k-2(p-1)} \rceil p$ pour tout $k \geq 2(p-1)$.*

Démonstration : Soit $K_{p \times q}$ une grille et k un entier $\geq 2(p-1)$. Posons $t = \lceil \frac{q-2(p-1)}{k-2(p-1)} \rceil$. Soit A_0 l'ensemble des $p-1$ premières colonnes de $K_{p \times q}$ et A_{t+1} les $p-1$ dernières colonnes de la grille $K_{p \times q}$.

Posons $A_1, A_2, \dots, A_{t-1}, A_t$ les sous-grilles de $K_{p \times q}$ telles que A_1 est engendré par les $k-2(p-1)$ premières colonnes de $K_{p \times q} \setminus \{A_0 \cup A_{t+1}\}$, A_2 est engendré par les $k-2(p-1)$ premières colonnes de $K_{p \times q} \setminus \{A_0 \cup A_1 \cup A_{t+1}\}$, $\dots$ A_{t-1} est engendré par les $k-2(p-1)$ premières colonnes de $K_{p \times q} \setminus \{A_0 \cup A_1 \dots A_{t-2} \cup A_{t+1}\}$. Posons, $A_t = K_{p \times q} \setminus \{A_0 \cup A_1 \dots \cup A_{t-1} \cup A_{t+1}\}$.

Grâce à la procédure **ConstruireGRID** ci-dessous, nous allons construire une grille G de taille $k \times \lceil \frac{q-2(p-1)}{k-2(p-1)} \rceil p$ à partir des sous-grilles $A_0, A_1, \dots, A_{t+1}$.

Construire $GRID(A_0, A_1, \dots, A_{t+1})$

Etape 1 : Posons q_t le nombre de colonnes de la sous-grille A_t . Si $q_t < k - 2(p - 1)$, alors : poser A_+ une grille de taille $p \times k - 2(p - 1) - q_t$. Pour tout $i, 1 \leq i \leq p$ relier tout sommet x_{i,q_t} de A_t au sommet $y_{i,1}$ de la grille A_+ . Soit A_t la grille résultante. Cette nouvelle grille A_t est de taille $p \times k - 2(p - 1)$.

Etape 2 : Pour tout $i, 1 \leq i \leq t - 1$: si i est paire, alors pour tout $j, 1 \leq j \leq k - 2(p - 1)$ relier tout sommet $x_{1,j}$ de A_{i+1} au sommet $y_{1,k-2(p-1)-j+1}$ de la grille A_i . Sinon, alors pour tout $j, 1 \leq j \leq k - 2(p - 1)$ relier tout sommet $x_{p,j}$ de A_{i+1} au sommet $y_{1,k-2(p-1)-j+1}$ de la grille A_i . La grille G ainsi obtenue est de taille $tp \times k - 2(p - 1)$.

Etape 3 : Si t est paire, alors construire une grille A_{t+2} de taille $p(t - 2) \times p - 1$ et faire : pour tout $j, 1 \leq j \leq p - 1$ relier tout sommet $x_{1,j}$ de la ligne 1 et la colonne j de A_{t+2} au sommet $y_{p,j}$ de la ligne p et la colonne j de A_0 . Pour tout $j, 1 \leq j \leq p - 1$, relier tout sommet $x_{p(t-2),j}$ de la ligne $p(t - 2)$ et la colonne j de A_{t+2} au sommet $y_{1,j}$ de la ligne 1 et la colonne j de A_{t+1} . Soit A_{t+3} cette nouvelle grille ; elle est de taille $tp \times p - 1$. Pour $j, 1 \leq j \leq tp$, relier tout sommet $x_{j,p-1}$ de la ligne j et la colonne $p - 1$ de A_{t+3} au sommet $y_{j,1}$ de G . Soit A_{t+4} une grille de taille $tp \times p - 1$. Pour tout $j, 1 \leq j \leq tp$, relier tout sommet $x_{j,1}$ de A_{t+4} au sommet $y_{j,k-(p-1)}$ de G . La grille G ainsi obtenue est de taille $pt \times k$.

Etape 4 : Sinon, construire une grille A_{t+2} de taille $p(t - 1) \times p - 1$. Pour tout $j, 1 \leq j \leq p - 1$ relier tout sommet $x_{1,j}$ de la ligne 1 et la colonne j de A_{t+2} au sommet $y_{p-1,j}$ de la ligne $p - 1$ et la colonne j de A_0 . Soit A_{t+3} cette nouvelle grille obtenue ; elle est de taille $tp \times p$.

Pour $j, 1 \leq j \leq tp$, relier tout sommet $x_{j,p-1}$ de la ligne j et la colonne $p - 1$ de A_{t+3} au sommet $y_{j,1}$ de G . Soit G la grille résultante. Soit A_{t+4} une grille de taille $(t - 1)p \times p - 1$. Pour $j, 1 \leq j \leq p - 1$ relier tout sommet $x_{1,j}$ de A_{t+1} au sommet $y_{(t-1)p,j}$ de la ligne $(t - 1)p$ et la colonne j de la grille A_{t+4} et soit A_{t+4} la grille résultante. A_{t+4} est donc une grille de taille $tp \times p - 1$. Pour tout $j, 1 \leq j \leq tp$, relier tout sommet $x_{j,1}$ de A_{t+4} au sommet $y_{j,k-(p-1)}$ de G . La grille G ainsi obtenue est de taille $pt \times k$. (cf. figures 3.16 et 3.17.)

Cette grille G formée est de taille $k \times \lceil \frac{q-2(p-1)}{k-2(p-1)} \rceil p$ et elle contient la grille $p \times q$. En effet, comme sur la figure 3.18, on peut relier chaque ligne L_j de A_i à la ligne L'_j de A_{i+1} par un chemin p_j de longueur au moins 1. Sur la figure 3.18, la zone contenant cette ensemble de chemins est coloré en noir. Ainsi, en réduisant chacun de ces chemins à une arête, on obtient une grille $p \times q$.

Nous avons par conséquent montré le lemme 3.18. $\square$

Sur la figure 3.16, les sous-grilles $A'_{t=9}$ et A'_{10} constituent le cas où la sous-grille A_9 a moins de 9 colonnes. Le nombre de colonnes de $A'_{t=9}$ est 3. Ainsi, la grille $K_{p \times q}$ aura $q = 79$ colonnes. Sur la grille $K_{p \times q}$ en (a), nous avons matérialisé la ligne 1 en bleu,

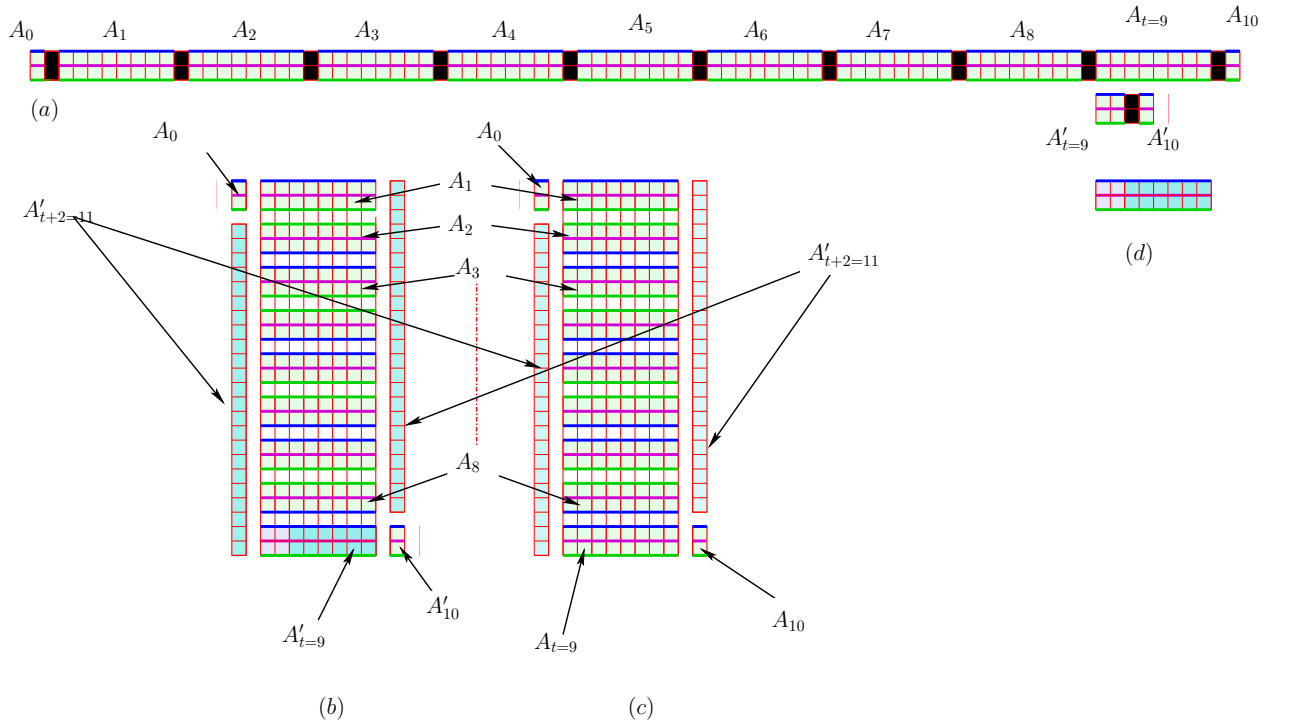


FIG. 3.16: (a) Une exemple de grille $K_{p \times q}$ avec $p = 3$ et $q = 85$ colonnes et une subdivision de la grille en sous-grilles $A_0, A_1, \dots, A_{t=9}, A_{t+1=10}$ telles que A_0 et A_{10} sont de taille 3×2 et les autres sous-grilles sont de taille 3×9 .

la ligne 2 en magenta et la ligne 3 en vert. Ces couleurs seront conservées par toutes les sous-grilles formées. En (d) on a le résultat de l'étape 1 sur la sous-grille $A'_{t=9}$. Dans cette étape, on augmente le nombre de colonnes de $A'_{t=9}$ jusqu'à 9. En (b) nous avons le résultat de l'étape 2 sur l'ensemble des sous-grilles $A_0, A_1, \dots, A_{t=9}, A_{t+1=10}$. La figure (b) représente le cas où $q = 79$. En (c), nous avons le résultat de l'étape 2 sur l'ensemble des sous-grilles $A_0, A_1, \dots, A_{t=9}, A_{t+1=10}$. La figure (c) représente le cas où $q = 85$. Nous pouvons aussi remarquer que $t = 8$; donc t est pair. C'est donc l'étape 4 qui sera exécuté. La différence entre les deux cas est l'emplacement de la sous-grille A_{t+1} . Dans la cas ou t est impair, on a A_{t+1} qui se place dans les $p - 1$ première colonnes de la grille G . Sur les figure (b) et (c), nous avons aussi matérialisé les sous-grilles A_{t+2} et A_{t+3} créées dans l'étape 4.

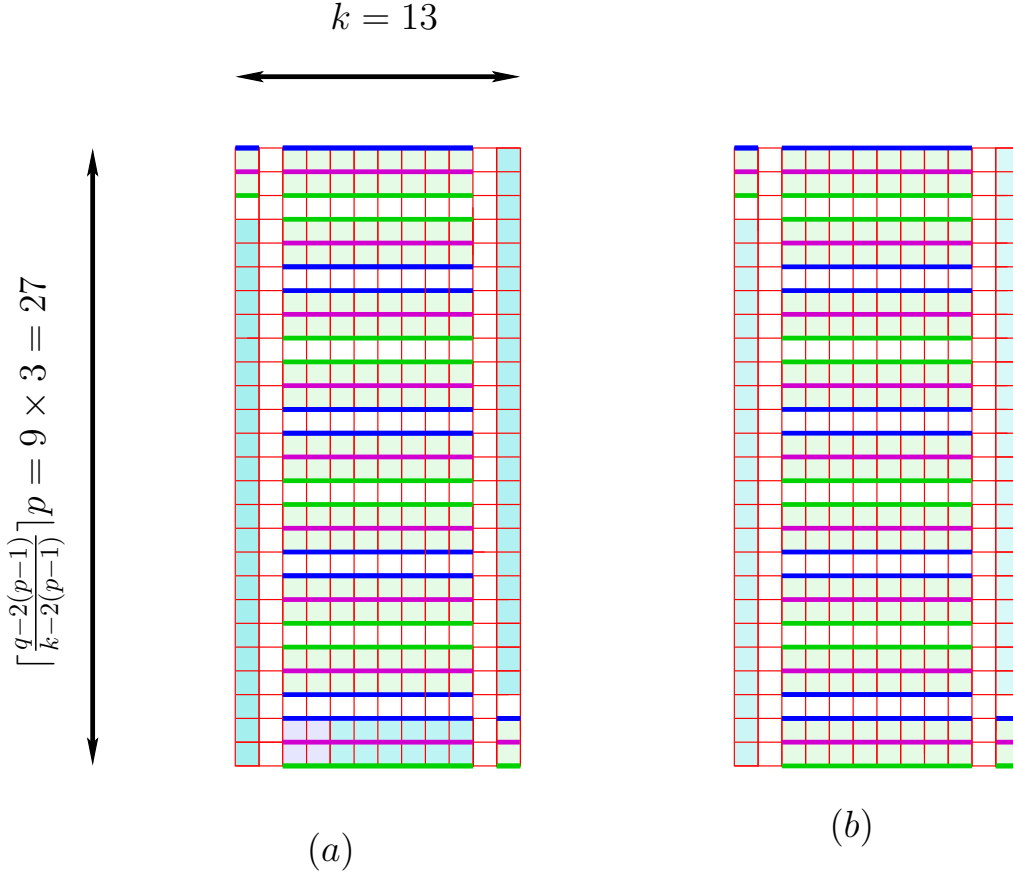


FIG. 3.17: En (a) et (b) on représente les grilles obtenues après l'étape 4. Les deux grilles sont identiques quelque soit le nombre de colonnes de la sous-grille A_t . On distingue grâce aux couleurs les lignes de la grille $K_{p \times q}$.

D'après le lemme 3.18, pour tout entier $k > 2(p-1)$, il existe une grille de taille $k \times \lceil \frac{q''' - 2(p-1)}{k-2(p-1)} \rceil p$ qui contient H''' . En choisissant k tel que $k = \lceil \frac{q''' - 2(p-1)}{k-2(p-1)} \rceil p$, alors on obtient une grille carrée $K_{k \times k}$.

En effet, si $k = \lceil \frac{q''' - 2(p-1)}{k-2(p-1)} \rceil p$, alors on a $k^2 - 2(p-1)kpq''' + 2p^2 - 2p = 0$.

Avec $\Delta = b^2 - 4ac$, on a $\Delta = 4(p-1)^2 - 4(-pq''' + 2p^2 - 2p) = 4(p^2 - 2p + 1 + pq''' - 2p^2 + 2p) = 4(pq''' + 1 - p^2) \geq 0$ car, par hypothèse on a $p < q'''$ ce qui veut dire que $pq''' > p^2$.

Donc $k = p - 1 + \sqrt{pq''' - p^2 + 1}$. Or, on a $q''' \leq 12p^2q$. Par conséquent, on a $k \leq p - 1 + \sqrt{12p^3q - p^2 + 1} = p - 1 + 2\sqrt{3p^3q - p^2 + 1} \leq p - 1 + 2\sqrt{3p^3q + 1} \leq$. Or on a $p \geq 1$ et $q \geq p$. Donc, on a $p - 1 + 2\sqrt{3p^3q + 1} \leq p - 1 + 2\sqrt{4p^3q}$. D'où, on a $k \leq 4p^{3/2}\sqrt{q} + p - 1$. Nous avons par conséquent montré le théorème 3.14.

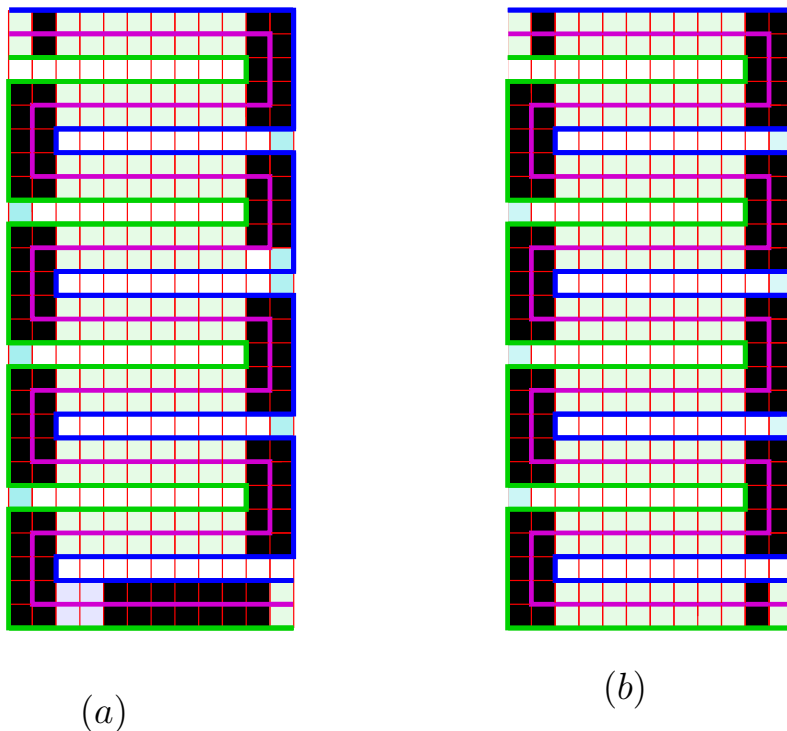


FIG. 3.18: Reconstruction du mineur $K_{p \times q}$ à partir des grilles (a) et (b) de la figure 3.17. Les zones colorées en noir contiennent les arêtes n'appartenant pas à $K_{p \times q}$. On remarque que pour toute sous-grille A_i et tout couple de lignes L_j, L_k de A_i , il existe une zone noir sur la grille $K_{p \times q}$ entre A_i et A_{i+1} telle que cette zone possède deux chemins p_j, p_k disjoints reliant respectivement les lignes L_j, L_k aux lignes L'_j, L'_k de A_{i+1} . Chaque ligne étant matérialisée par une couleur, il est facile de faire la vérification.

3.4.2 Conclusion

Dans ce chapitre nous avons présenté un algorithme de complexité linéaire qui pour tout graphe 2-connexe soit exhibe un mineur $K_{2,4}$, soit trouve deux sommets u, v dont leur suppression engendre un graphe planaire-extérieur. Nous avons par conséquent montré que pour tout graphe sans mineur $K_{2,4}$, nous pouvons trouver en temps linéaire deux sommets u, v dont leur suppression engendre un graphe planaire-extérieur. Ce résultat permet donc de dire que les graphes sans mineur $K_{2,4}$ sont de largeur arborescente au plus 4. Cette borne optimale améliore ainsi le résultat de Bodlaender *et al.* qui est de 6. Nous montrons aussi que ce résultat ne peut pas être

étendu trivialement à tout graphe planaire sans mineur $K_{2,r}$ en montrant que pour tout entier k , il existe un graphe planaire 2-connexe sans mineur $K_{2,5}$ pour lequel il faut enlever k sommets pour le rendre planaire-extérieur.

De plus, vu que $K_{2,4}$ possède 8 arêtes par rapport aux 15 arêtes de K_6 , alors la question qu'on pourrait se poser est de savoir ce qu'on peut dire sur la structure des graphes sans un mineur H ayant plus d'arêtes par exemple avec $H = K_{2,4} \cup \{e\}$?

Que pouvons nous dire de la structure des graphes sans mineur $K_{2,r}$ pour $r \geq 5$?

Par ailleurs, nous avons montré que pour tout graphe H dessinable sur une grille $p \times q$ et tout graphe G sans mineur H , on a la largeur arborescente de G qui est au plus $O(p^{3/2}\sqrt{q})$. Ainsi donc nous montrons que les graphes sans mineur $K_{2,r}$ sont de largeur arborescente $O(\sqrt{r})$. Ce qui améliore la borne actuelle due à Thilikos, qui est de $r + 2$. Nous montrons également que ce résultat est asymptotiquement optimal.

La question qu'on pourrait se poser est de savoir si ce résultat pourrait être étendu à d'autres familles de graphes plus larges.

Chapitre 4

Schémas de routage compacts de plus courts chemins

4.1 Introduction

Dans toute cette partie nous nous intéressons au routage dans les réseaux interconnectant des machines par des liens de communication bi-directionnels.

La communication point-à-point dans les réseaux d'ordinateurs se fait grâce à une fonction de routage. Cette fonction de routage peut être simple pour des réseaux de taille réduite. Cependant dans les réseaux de grande taille il devient important de réduire la quantité d'informations que chaque nœud doit mettre à disposition de la fonction de routage pour permettre la communication. De même il devient capital d'acheminer les messages par des chemins aussi courts que possible.

4.1.1 Caractéristiques d'un schéma de routage

Les composants d'un *schéma de routage* sont les tables de routage, les adresses, les en-têtes et un algorithme de routage. La *fonction de routage* prend en paramètre l'en-tête de message arrivant et, en fonction de la table de routage du sommet courant, elle détermine le numéro de port par lequel le message doit être retransmis. L'en-tête de message est généré par le sommet émetteur et reste souvent fixe tout au long du trajet. Elle est générée à partir de sa table de routage et de l'adresse de destination. Dans beaucoup de schémas, l'en-tête se résume à l'adresse de destination.

L'enjeu est donc d'optimiser non seulement la longueur des routes mais aussi l'espace requis par les tables de routage, la taille des en-têtes et des adresses, de même que la latence dans chaque routeur.

De nombreux travaux ont été consacrés à la conception de schéma de routage comportant un compromis taille des tables / longueur des routes en passant. Dans le cas d'une topologie quelconque, les tables sont de taille $n^{\Theta(1/s)}$ où s est le *facteur d'étirement*, le ratio maximum sur toutes les paires de nœuds (x, y) entre la longueur de la route et la distance entre x et y . (Voir par exemple [AGM06, AGM+08, Cow01, PU89, TZ01] et [Dom07, KCFB07, Gav01] pour une introduction au routage compact dans l'Internet). Dans toutes ces solutions, le passage à l'échelle s'avère coûteux lorsque s est très proche de 1, c'est-à-dire lorsque les

routes sont presque des plus courts chemins, la taille des tables étant alors polynomiale en la taille du réseau.

En passant par l'étude structurelle fine de la topologie sous-jacente du réseau, il est possible d'obtenir dans certains cas des schémas de plus courts chemins avec des tables de taille *poly-logarithmique* en la taille du réseau. C'est le cas des topologies arborescentes où il a été montré que des tables de $O(\log n)$ bits sont possibles [FG01b, TZ01]. Notons que ce résultat est à la base des schémas plus généraux de [TZ01] et de [AGM06, AGM⁺08]. Il est également connu que les réseaux de largeur arborescente au plus t (une généralisation des arbres) possèdent un routage de plus courts chemins avec des tables, adresses et en-têtes de $O(t \log^2 n)$ bits [Pel00]. Pour $t = 1$ (c'est-à-dire le cas des arbres), cette borne n'est pas optimale, et il est toujours ouvert de savoir si $\Theta(\log n)$ bits suffisent pour $t = 2$.

4.1.2 Présentation des résultats

Dans cette partie, nous allons présenter très brièvement les principaux résultats de ce chapitre. Dans un premier temps, nous allons nous intéresser aux routages dans les graphes planaire-extérieurs.

Nous montrons le résultat suivant :

Théorème 4.1 *Tout graphe planaire-extérieur connexe et non valué à n sommets admet un schéma de routage de plus courts chemins avec des adresses, des entêtes et des tables de routage $O(\log n)$ bits. De plus le temps de routage (latence) est constant.*

Ce résultat est une généralisation du schéma de routage pour les arbres, présenté par Thorup et Zwick [TZ01] et indépendamment par Fraigniaud et Gavoille [FG01b]. Ce résultat, est assez simple à comprendre, mais nous avons choisi de le présenter ici pour expliquer principe qui est le même que celui qui est utilisé dans le résultat principal présenté dans la deuxième partie du chapitre. La deuxième partie de ce chapitre sera consacrée essentiellement à l'extension de ce résultat à des graphes beaucoup plus généraux.

Dans un premier temps, nous donnons une généralisation aux graphes valués (k, r) -cellulaires. Les graphes (k, r) -cellulaires sont des graphes ne contenant pas $K_{2,r}$ comme mineur et dont les composantes 2-connexes peuvent être rendues planaire-extérieures par la suppression de k sommets. Par exemple, les réseaux $(2,4)$ -cellulaires contiennent les arbres, les graphes planaire-extérieurs et plus généralement tous les graphes sans mineur $K_{2,4}$ qui contiennent, par exemple des arbres de K_5 ou de $K_{3,3}$ qui sont de genre non borné. Plus précisément, nous montrons le résultat suivant :

Théorème 4.2 *Tout graphe (k, r) -cellulaire connexe, valué à n sommets admet un schéma de routage de plus court chemins avec des adresses et des en-têtes de $O(\log r \log n)$ bits et des tables de routage de $O(kr \log n)$ bits. Le temps de décision est de $O(\log r)$.*

Notons que le théorème 4.1 est un corollaire du théorème 4.2. De plus, vu que les graphes sans mineur $K_{2,4}$ sont $(2, 4)$ -cellulaires, alors on a le corollaire suivant :

Corollaire 4.3 *Tout graphe sans mineur $K_{2,4}$, connexe et valué à n sommets, admet un schéma de routage de plus courts chemins avec des tables, des adresses et des en-têtes de message de taille $O(\log n)$ bits. Le temps de routage est constant.*

Dans un deuxième temps, nous étendons le routage dans les graphes planaire-extérieurs à une autre famille de graphes plus générale, les graphes t -feuillet. Un graphe t -feuillet G est un graphe dont les arêtes peuvent être partitionnées en un ensemble de sous-graphes de G planaire-extérieurs tels que chaque arête de G soit contenue dans au moins un sous-graphe et il y a au plus t sommets appartenant à la fois à plusieurs sous-graphes. Par exemple, les graphes planaire-extérieurs sont 0-feuillet. Plus précisément, nous montrons le résultat suivant :

Théorème 4.4 *Tout graphe t -feuillet connexe, non valué possède un schéma de routage de plus courts chemins utilisant des tables de $O(t \log n)$ bits par sommets, des adresses et des en-têtes de $O(\log n)$ bits. Le temps de décision est de $O(\log t)$ par sommet.*

Le théorème 4.4 s'étend au modèle *ports fixés*, c'est-à-dire le cas où les numéros de ports sont fixés avant la construction des tables et des adresses, et ne peuvent être changés. Alors, les termes « $\log n$ » de l'énoncé précédent devant être remplacés par « $\log^2 n / \log \log n$ ».

4.2 Routage compact de plus courts chemins dans les graphes planaire-extérieurs non valués

Un graphe planaire-extérieur ne contient pas de mineur $K_{2,3}$. Donc, un graphe planaire-extérieur est un graphe $(0, 3)$ -cellulaire. Donc, le théorème 4.1 comme nous l'avons dit précédemment est un corollaire du théorème .

Cette partie sera consacrée essentiellement à la démonstration du théorème 4.1. L'objectif est d'expliquer le principe qui sera utilisé dans le cas général, c'est-à-dire le théorème 4.2.

4.2.1 Construction du schéma de routage pour les graphes planaire-extérieurs non valués

Soit G un graphe planaire-extérieur d'ordre n et $\tilde{G}$ un plongement planaire-extérieur de G .

Soit une numérotation des sommets de G suivant un parcours bordant la face extérieure de $\tilde{G}$ par des entiers consécutifs de $[1, \dots, n]$.

Soit $\mathcal{T}$ un arbre couvrant G de plus courts chemins, enraciné en un sommet r choisi arbitrairement.

Pour tout sommet $x \in G$:

- soit $\mathcal{T}_x$ le sous-arbre de $\mathcal{T}$ enraciné en x et $H_x = G \setminus \mathcal{T}_x$, le sous-graphe induit par les sommets de $G \setminus \mathcal{T}_x$.

- soit $\Omega(x)$ l'ensemble des sommets $v \in N(x) \cap \mathcal{T}_x$ tels que $\exists y \in H_x$ et v se trouve sur un plus court chemin dans G entre x et y .
- soit T_x^H un arbre de plus courts chemins dans G enraciné en x , couvrant H_x .
- $TABLE\text{-}\mathcal{T}_x$ une table de routage construite à partir de l'arbre $\mathcal{T}$ dans laquelle seuls les descendants de x dans $\mathcal{T}$ sont préservés. Ainsi, pour toute destination y , $TABLE\text{-}\mathcal{T}_x[y]$ donne le numéro de port de l'arête menant vers y si $y \in \mathcal{T}_x$.
- $TABLE\text{-}H_x$ une table de routage construite à partir de l'arbre T_x^H en compactant sous forme de segments, l'ensemble des destinations de T_x^H utilisant le même port. Ainsi, pour toute destination y , $TABLE\text{-}H_x[y]$ donne le numéro de port de l'arête menant vers y si $y \in H_x$.

4.2.1.1 L'algorithme de routage

L'algorithme de routage présenté ci-dessous est élémentaire, l'originalité venant de son implémentation.

ROUTE-(y) :

1. **Si** $y \in \mathcal{T}_x$, alors retourner $TABLE\text{-}\mathcal{T}_x[y]$,
2. **Sinon**, retourner $TABLE\text{-}H_x[y]$

4.2.1.2 Explications de l'algorithme

Nous pouvons remarquer sans ambiguïté que tout sommet $y \in G$ est soit dans $\mathcal{T}_x$ soit dans H_x .

En outre, si $y \in \mathcal{T}_x$, alors le port permettant d'aller en y par de plus courts chemins dans G est dans la table $TABLE\text{-}\mathcal{T}_x$. Donc, en retournant $TABLE\text{-}\mathcal{T}_x[y]$, on retourne un port sur un plus court chemin entre x et y dans G .

Si $y \notin \mathcal{T}_x$, alors y est dans H_x et dans ce cas, le port permettant d'aller en y par de plus courts chemins dans G est dans la table $TABLE\text{-}H_x$. Donc en retournant $TABLE\text{-}H_x[y]$ on retourne un port sur un plus court chemin entre x et y .

L'algorithme *ROUTE* est donc un algorithme de routage de plus courts chemins.

4.2.2 Implémentation

4.2.2.1 Les tables de routage

Pour chaque sommet $x \in G$, la table de routage de x est constituée par l'union des tables $TABLE\text{-}\mathcal{T}_x$ et $TABLE\text{-}H_x$.

La table de routage $TABLE\text{-}\mathcal{T}_x$ est construite à partir de l'arbre $\mathcal{T}$ dans laquelle seuls les sommets descendants de x dans $\mathcal{T}$ sont préservés. D'après [FG01b] et [TZ01], nous pouvons implémenter $TABLE\text{-}\mathcal{T}_x$ en $O(\log n)$ bits. La technique utilisée ici est de partager l'information entre x et les sommets de $\mathcal{T}_x$.

La table de routage $TABLE\text{-}H_x$ est construite en compactant sous forme de segments l'ensemble des destinations de H_x passant par le même port. En d'autres termes, pour chaque port p de sortie, il existe une entrée unique dans $TABLE\text{-}H_x$

qui contient l'ensemble des segments de sommets passant par p . De plus, on a le théorème suivant :

Théorème 4.5 [FJ88] *Si un graphe planaire-extérieur G est tel que les sommets sont numérotés par des entiers consécutifs suivant un parcours bordant la face extérieure d'un plongement planaire-extérieur de G alors, pour tout sommet x , et tout voisin v de x , l'ensemble des destinations telles que le plus court chemin dans G passe par v peut être compacté en un segment d'entiers modulo $|G|$.*

Par conséquent, chaque entrée de la table $TABLE-H_x$ possède au plus un segment. Soit $\Omega(x) \subseteq N(x)$ minimum par taille tel que $\forall y \in H_x, \exists v \in \Omega(x)$ tel que v est sur un plus court chemin entre x et y dans G .

Alors, la table $TABLE-H_x$ possède $|\Omega(x)|$ entrées. Grâce au lemme suivant, nous pouvons estimer sa taille.

Lemme 4.6 *Pour tout sommet $x \in G$, on a $|\Omega(x)| \leq 6$.*

Démonstration : L'ensemble $\Omega(x)$ peut être partitionné en deux sous-ensembles : $\Omega_1(x)$ et $\Omega_2(x)$.

Soit $\Omega_2(x)$ le voisinage de x dans H_x .

Soit $\Omega_1(x)$ un ensemble minimum de voisins v de x dans $\mathcal{T}$ tels que $\exists y \in H_x$ et v est sur un plus court chemin de x à y et tout chemin entre x et y qui utilise un sommet de $\Omega_2(x)$ n'est pas un plus court chemin dans G .

Alors, on a $|\Omega(x)| \leq |\Omega_1(x)| + |\Omega_2(x)|$.

Nous allons montrer que $|\Omega_1(x)| + |\Omega_2(x)| \leq 6$. Mais auparavant, nous avons besoin de montrer le lemme suivant :

Lemme 4.7 *Pour tout sommet $x \in G$, on a $|\Omega_1(x)| \leq 2$ et $|\Omega_2(x)| \leq 6$.*

Démonstration : Montrons par l'absurde que $|\Omega_1(x)| \leq 2$. En effet, supposons que $|\Omega_1(x)| \geq 3$. Donc $\exists X \subseteq H_x$ et au moins 3 sommets $u, v, w \in N(x) \cap \mathcal{T}_x$ tel que p_1, p_2, p_3 sont 3 chemins dans G entre x et X tel que $u \in p_1, v \in p_2$ et $w \in p_3$. De plus, par le fait que $u, v, w \notin H_x$, alors p_1, p_2, p_3 sont des chemins disjoints de longueur d'au moins 2. Par ailleurs, vu que G est connexe et que $X \not\subseteq \mathcal{T}_x$, alors il existe un ensemble P de chemins dans $\mathcal{T}$ entre r et chaque sommet de X et $\forall p \in P, x \notin p$. Donc, considérant le sous-graphe de G induit par les sommets de $r \cup P \cup X \cup \{x\} \cup p_1 \cup p_2 \cup p_3$ et en contractant $r \cup P$, on obtient un mineur $K_{2,3}$; ce qui est absurde car G est sans mineur $K_{2,3}$. Donc, pour tout $x \in G$, on a $|\Omega_1(x)| \leq 2$.

Pour montrer que $|\Omega_2(x)| \leq 6$, il suffit juste de montrer le lemme suivant :

Lemme 4.8 *Si $x \in G$ est tel que $\text{dist}_G(r, x) = k$, alors :*

1. *pour tout sommet $v \in \Omega_2(x)$, on a $k - 1 \leq \text{dist}_G(r, v) \leq k + 1$.*
2. *pour tout $k - 1 \leq i \leq k + 1$, il y a au plus deux sommets $v \in \Omega_2(x)$ tels que $\text{dist}_G(r, v) = i$.*

Démonstration : Nous allons faire une preuve par l'absurde de 1 en considérant 2 cas : le cas où on suppose $v \in \Omega_2(x)$ tel que $\text{dist}_G(r, v) \leq k - 2$ et le cas où on suppose $v \in \Omega_2(x)$ tel que $\text{dist}_G(r, v) \geq k + 2$.

CAS 1 : Supposons que $\exists v \in \Omega_2(x)$ tel que $\text{dist}_G(r, v) \leq k - 2$. Soit $p_{r,v}$ un plus court chemin dans G entre r et v . Vu que v est adjacent à x dans G , alors il existe un chemin $p_{r,x} = p_{r,v} \cup (v, x)$ entre r et x de longueur $\leq k - 1$ ce qui est absurde vu que par hypothèse on a $\text{dist}_G(r, x) = k$. Donc, il n'existe pas de sommet v adjacent à x dans G tel que $\text{dist}_G(r, v) \leq k - 2$.

CAS 2 : Supposons que $\exists v \in \Omega_2(x)$ tel que $\text{dist}_G(r, v) \geq k + 2$. Soit $p_{r,x}$ un plus court chemin dans G entre r et x . Vu que x est adjacent à v dans G , alors il existe un chemin $p_{r,v} = p_{r,x} \cup (x, v)$ entre r et v de longueur $k + 1$ ce qui est absurde vu que par hypothèse on a $\text{dist}_G(r, v) \geq k + 2$. Donc, il n'existe pas de sommet v adjacent à x dans G tel que $\text{dist}_G(r, v) \geq k + 2$. Par conséquent, tout sommet $v \in \Omega_2(x)$, est tel que $k - 1 \leq \text{dist}_G(r, v) \leq k + 1$. Ce qui termine la preuve du point 1 du lemme 4.8.

Montrons maintenant, le point 2 du lemme 4.8. En effet, supposons que u, v, w soient 3 sommets de $\Omega_2(x)$, tels que $\exists i, k - 1 \leq i \leq k + 1$ et $\text{dist}_G(r, u) = \text{dist}_G(r, v) = \text{dist}_G(r, w) = i$. Donc, il existe 3 chemins $p_{r,u}, p_{r,v}, p_{r,w}$ dans $\mathcal{T}$ tels que $p_{r,u}$ est le chemin de $\mathcal{T}$ entre r et u , $p_{r,v}$ est le chemin de $\mathcal{T}$ entre r et v et $p_{r,w}$ est le chemin de $\mathcal{T}$ entre r et w . Soit H le sous-graphe de G induit par, $p_{r,u} \cup (u, x) \cup p_{r,v} \cup (v, x) \cup p_{r,w} \cup (w, x)$. Soit a_1 le plus proche ancêtre commun entre u et v et a_2 le plus proche ancêtre commun dans $\mathcal{T}$ entre u et w . Contracter dans H le chemin dans H entre r et a_1 et le chemin entre a_1 et a_2 et soit r_c le nouveau sommet obtenu. Vu que dans G , les sommets u, v, w sont tous à la même distance par rapport à r , alors il existe 3 chemins $p_{r_c,u}, p_{r_c,v}, p_{r_c,w}$ dans H tels que $p_{r_c,u}$ est le chemin dans H entre r_c et u , $p_{r_c,v}$ est le chemin de H entre r_c et v et $p_{r_c,w}$ le chemin de H entre r_c et w . De plus, $|p_{r_c,u}| \geq 1, |p_{r_c,v}| \geq 1, |p_{r_c,w}| \geq 1$ car, sinon on aurait au moins deux sommets $x, y \in \{u, v, w\}$ tels que x est sur un plus court chemin dans G entre r et y ; ce qui est une contradiction avec l'hypothèse. Par conséquent, en considérant les chemins $P_1 = p_{r_c,u} \cup (u, x), P_2 = p_{r_c,v} \cup (v, x)$ et $P_3 = p_{r_c,w} \cup (w, x)$ on obtient trois chemins de longueur au moins 2 entre r_c et x . Nous avons donc construit mineur $K_{2,3}$ dans G ; ce qui est absurde par hypothèse.

D'où, pour tout $i, k - 1 \leq i \leq k + 1$, il y a au plus deux sommets $v \in \Omega_2(x)$ tels que $\text{dist}_G(r, v) = i$. Ce qui termine la preuve du lemme 4.8. $\square$

Grâce au lemme 4.8, nous avons montré que $|\Omega_2(x)| \leq 6$. Donc, nous avons montré lemme 4.7. $\square$

Pour montrer que $|\Omega_1(x)| + |\Omega_2(x)| \leq 6$, nous allons montrer le lemme suivant :

Lemme 4.9 1 Si x admet 2 voisins $u, v \in \Omega_2(x)$ tels que $\text{dist}_G(r, u) = \text{dist}_G(r, v) = k + 1$, alors on a $\Omega_1(x) = \emptyset$.
2 Si x admet 2 voisins dans $\Omega_1(x)$, alors il n'existe pas de sommet $w \in \Omega_2(x)$ tel que $\text{dist}_G(r, w) = k + 1$.

Démonstration : Montrons le point 1 du lemme 4.9 par l'absurde. En effet, soit x un sommet de G tel que $\exists u, v \in \Omega_2(x)$, $\text{dist}_G(r, u) = k + 1$ et $\text{dist}_G(r, v) = k + 1$.

Supposons qu'il existe un sommet w tel que $w \in \Omega_1(x)$. Posons w_1 un sommet de H_x tel que $\text{dist}_{H_x}(x, w_1) > \text{dist}_G(x, w_1)$ et w_1 utilise w . Posons p_{r,w_1} le chemin dans $\mathcal{T}$ entre r et w_1 , $p_{r,u}$ le chemin dans $\mathcal{T}$ entre r et u et, $p_{r,v}$ le chemin dans $\mathcal{T}$

entre r et v . Vu que $x \neq r$, on a $k \geq 1$. Donc $|p_{r,v}| \geq 2$ et $|p_{r,w}| \geq 2$. Posons p_{x,w_1} un plus court chemin dans G entre x et w_1 passant par w .

Si p_{r,w_1} ne passe ni par u ni par v , alors les chemins $p_{r,u} \cup (u, x)$, $p_{r,v} \cup (v, x)$, et $p_{r,w_1} \cup p_{x,w_1}$ constituent 3 chemins disjoints de longueur ≥ 2 entre r et x . Donc G admet $K_{2,3}$ comme mineur ; ce qui est absurde par hypothèse. Donc, p_{r,w_1} passe soit par u soit par v .

On peut donc supposer sans perdre de généralité que p_{r,w_1} passe par u . Donc, on a $|p_{r,w_1}| = |p_{r,u}| + |p_{u,w_1}|$ avec p_{u,w_1} un plus court chemin entre u et w_1 . Or $|p_{r,u}| = k + 1$. Donc $|p_{r,w_1}| = k + 1 + |p_{u,w_1}|$. De plus, par définition de $\Omega_1(x)$ et $\Omega_2(x)$ tout plus court chemin p dans H_x entre x et w_1 est tel que $p > p_{x,w_1}$. Donc, $|p_{u,w_1}| + 1 > p_{x,w_1}$. D'où, $|p_{u,w_1}| + 1 + k > p_{x,w_1} + k$. Or $|p_{r,w_1}| = k + 1 + |p_{u,w_1}|$. Donc on a $|p_{r,w_1}| > p_{x,w_1} + k$; ce qui est absurde car p_{r,w_1} est un plus court chemin dans G entre r et w_1 . Par conséquent, il n'existe pas de sommet w tel que $w \in \Omega_1(x)$.

Montrons le point 2 du lemme 4.9 par l'absurde.

Soit x un sommet de G tel que $\exists u, v \in \Omega_1(x)$. Supposons qu'il existe un sommet w tel que $w \in \Omega_2(x)$. Soit u_1 un sommet de H_x tel que $\text{dist}_{H_x}(x, u_1) > \text{dist}_G(x, u_1)$ et u_1 utilise u . Soit v_1 un sommet de H_x tel que $\text{dist}_{H_x}(x, v_1) > \text{dist}_G(x, v_1)$ et v_1 utilise v .

Posons p_{r,u_1} le chemin dans $\mathcal{T}$ entre r et u_1 , et p_{x,u_1} un plus court chemin dans G entre x et u_1 tel que $u \in p_{x,u_1}$. Posons p_{r,v_1} le chemin dans $\mathcal{T}$ entre r et v_1 et p_{x,v_1} un plus court chemin dans G entre x et v_1 tel que $v \in p_{x,v_1}$.

Vu que $x \neq r$, on a $k \geq 1$. Donc le chemin $p_{r,w}$ dans $\mathcal{T}$ entre r et w est de longueur au moins 2.

Si $w \notin p_{r,u_1}$ et $w \notin p_{r,v_1}$, alors les chemins $p_{r,u_1} \cup p_{x,u_1}$, $p_{r,v_1} \cup p_{x,v_1}$ et $p_{r,w} \cup (w, x)$ constituent 3 chemins disjoints entre r et x . On en déduit un mineur $K_{2,3}$ dans G ; ce qui est absurde par hypothèse. Donc, on a soit $w \in p_{r,u_1}$ soit $w \in p_{r,v_1}$.

On peut donc supposer sans perdre de généralité que $w \in p_{r,u_1}$. On a donc, $|p_{r,u_1}| = k + 1 + |p_{w,u_1}|$ avec p_{w,u_1} un plus court chemin dans G entre w et u_1 . En outre, par définition de $\Omega_1(x)$ et $\Omega_2(x)$, on a $|p_{w,u_1}| + 1 > |p_{x,u_1}|$. Donc, $|p_{w,u_1}| + 1 + k > |p_{x,u_1}| + k$; ce qui est absurde car $|p_{w,u_1}| + 1 + k$ est la longueur du chemin dans $\mathcal{T}$ entre r et u et $\mathcal{T}$ est un arbre de plus courts chemins dans G . Donc $w \notin p_{r,u_1}$.

Par conséquent, on a $\Omega_2(x) = \emptyset$. Ce qui termine la preuve du lemme 4.9. $\square$

Par conséquent, on peut facilement en déduire que $|\Omega(x)| \leq 6$. Pour cela, on considère chacun des cas de figure suivants :

CAS 1 : Si $|\Omega_1(x)| = 2$, alors on a $|\Omega_2(x)| \leq 4$. En effet, si on avait $|\Omega_2(x)| \geq 4$, alors il existerait au moins 3 sommets $u, v, w \in \Omega_2(x)$ tels que $\text{dist}_G(r, u) = \text{dist}_G(r, v) = \text{dist}_G(r, w)$; ce qui est impossible d'après le lemme 4.8. D'où on a $|\Omega(x)| \leq |\Omega_1(x)| + |\Omega_2(x)| = 6$.

CAS 2 : Si $|\Omega_2(x)| = 5$, alors il y a au moins un sommet $u \in \Omega_2(x)$ tel que $\text{dist}_G(r, u) = k + 1$ car sinon, il existerait au moins 3 sommets $u, v, w \in \Omega_2(x)$ tels que $\text{dist}_G(r, u) = \text{dist}_G(r, v) = \text{dist}_G(r, w)$; ce qui est impossible d'après le lemme 4.8. Donc, on a $|\Omega_1(x)| \leq 1$ d'après le cas 1. D'où, on a $|\Omega(x)| \leq |\Omega_1(x)| + |\Omega_2(x)| = 6$.

CAS 3 : Si $|\Omega_2(x)| = 6$, alors il y a deux sommets $u, v \in \Omega_2(x)$ tels que $\text{dist}_G(r, u) = \text{dist}_G(r, v) = k + 1$ car sinon, il existerait au moins 3 sommets $u, v, w \in \Omega_2(x)$ tels que $\text{dist}_G(r, u) = \text{dist}_G(r, v) = \text{dist}_G(r, w)$; ce qui est impossible d'après le lemme 4.8. D'où, on a $|\Omega_1(x)| = 0$ d'après le lemme 4.9. Donc, on a $|\Omega(x)| \leq |\Omega_1(x)| + |\Omega_2(x)| = 6$.

CAS 4 : Si $|\Omega_2(x)| \leq 4$, alors on a $|\Omega(x)| \leq |\Omega_1(x)| + |\Omega_2(x)| \leq 6$ car, d'après le lemme 4.7 on a $|\Omega_1(x)| \leq 2$. Donc, on a $|\Omega(x)| \leq |\Omega_1(x)| + |\Omega_2(x)| = 6$.

Donc, nous avons montré que pour tout x , $|\Omega(x)| \leq 6$. Ce qui termine la preuve du lemme 4.6. $\square$

Par conséquent, nous avons montré que la table $TABLE-H_x$ possède au plus 6 entrées et donc 6 segments. De plus, vu qu'un segment peut être codé en $2 \log n$ bits alors $TABLE-H_x$ peut être codée en $12 \log n$ bits.

Le temps de routage est le temps nécessaire pour trouver un segment dans un ensemble d'au plus 6 segments. Le temps de routage est donc constant.

4.2.2.2 Les adresses

L'adresse de chaque sommet x est constituée par son numéro issu de la numérotation suivant le bord de la face extérieur du plongement de G plus un numéro issu d'une numérotation utilisée pour le routage dans $\mathcal{T}_x$ de taille $\log n + o(\log n)$ bits [TZ01]. Donc l'adresse peut être codée pour chaque sommet en $2(\log n) + o(\log n)$ bits. Les en-têtes de message sont constituées par l'adresse du sommet de destination. Il sont donc de $2(\log n) + o(\log n)$.

4.3 Généralisations

4.3.1 Routage compact de plus courts chemins dans les graphes (k, r) -cellulaires

Nous venons de présenter un schéma de routage pour les graphes planaire-extérieurs non valués. Dans cette partie, nous allons utiliser le même principe pour généraliser ce résultat aux graphes valués, (k, r) -cellulaires.

Nous pouvons remarquer que la construction du schéma de routage pour les graphes planaire-extérieurs consiste à trouver une bonne partition des sommets du graphe et à exploiter les propriétés caractéristiques de chaque partie. Les deux propriétés fondamentales de ce schéma de routage sont celles énoncées dans le lemme 4.6 et le théorème 4.5. En effet, une généralisation naturelle de ce résultat consiste à trouver une généralisation de chacune de ces deux propriétés aux graphes (k, r) -cellulaires. Nous avons besoin de la définition suivante :

Pour tout ensemble X de sommets de G , on note $\text{seg}(X)$ le plus petit nombre de segments dont l'union est X .

Définition 4.10 (Propriété d'intervalle) *Un graphe G a une f -propriété d'intervalle s'il existe une numérotation des sommets de G telle que : pour tout sommet $x \in G$, tout $\Omega(x) \subseteq N(x)$ et tout arbre T couvrant G de plus courts chemins, enraciné en x , on a $\sum_{v \in \Omega(x)} \text{seg}(T_v) \leq f(|\Omega(x)|)$.*

Voici comment on peut reformuler le théorème 4.5 en fonction de la Propriété d'intervalle :

Théorème 4.11 *Si G est un graphe planaire-extérieur, G admet une f -propriété d'intervalle telle que $\forall k \in \mathbb{N}, f(k) = k$.*

4.3.1.1 Propriété d'intervalle

Le lemme suivant peut être vu comme une généralisation du théorème 4.11 :

Lemme 4.12 *Tout graphe (k, r) -cellulaire, 2-connexe a la f -propriété d'intervalle avec $f(d) \leq 2d + 5kr$.*

Démonstration : Soit G un graphe (k, r) -cellulaire bi-connexe.

Soit X le plus petit ensemble de sommets tels que $G \setminus X$ est planaire-extérieur.

Considérons la fonction de numérotation suivante qui attribue à chaque sommet de G un numéro unique compris entre 1 et $|G|$:

1. Numéroter en premier les sommets de X par des entiers consécutifs compris entre 1 et $|X|$.
2. Considérons un plongement planaire-extérieur de $G \setminus X$, puis numéroter consécutivement les sommets $G \setminus X$ suivant un parcours bordant la face extérieure de ce plongement.

Soit x un sommet de G et soit $\Omega(x) = \{v_1, v_2, \dots, v_t\}$ un ensemble de voisins de x dans G .

Soit T un arbre couvrant G de plus courts chemins, enraciné en x . Soient $T_{v_1}, T_{v_2}, \dots, T_{v_t}$ les sous-arbres de T enracinés respectivement en $v_1, v_2, \dots, v_t$.

Calculons $\sum_{v \in \Omega} \text{seg}(T_v)$.

Pour cela, nous allons d'abord calculer le nombre de segments qui utilisent au moins un sommet de X et ensuite le nombre de segments qui n'utilisent aucun sommet de X .

Nous allons distinguer deux types de segments parmi ceux utilisant X . D'une part, les segments qui sont adjacents à X dans T et d'autres part les segments qui ne sont adjacents à aucun sommet de X dans T . Un segment s est dit adjacent à X dans T s'il existe au moins un sommet $y \in s$ et un sommet $z \in X$ tel que $(y, z) \in T$.

Ainsi, pour tout sommet u de X tel que $u \neq x$, posons :

1. S_u l'ensemble des segments ayant u comme ancêtre dans T et dont chaque segment contient au moins un sommet adjacent à u dans T .

2. W_u l'ensemble des segments ayant u comme ancêtre dans T et qui ne contiennent aucun sommet adjacent à u dans T .

Calculons le nombre maximum de segments de S_u pour u quelconque dans X . Nous allons montrer que $|S_u| \leq 2r$. Pour cela, partitionnons les segments de S_u en deux groupes.

Soit S_u^1 l'ensemble des segments s de S_u tels que : $\exists v_1 \in s, v_2 \notin T_u$ et p_{v_1, v_2} un chemin dans G entre v_1 et v_2 tel que $p_{v_1, v_2} \cap S_u = \emptyset$.

Soit S_u^2 l'ensemble des segments s' de S_u tels que : $\exists v_1 \in s, v_2 \notin T_u$ et pour tout chemin p_{v_1, v_2} dans G entre v_1 et v_2 on a $p_{v_1, v_2} \cap S_u \neq \emptyset$.

Nous allons montrer par l'absurde que $|S_u^1| < r$ et $|S_u^2| < r$.

En effet, supposons que $|S_u^1| \geq r$. Par définition de S_u^1 , à partir de chacun des segments de S_u^1 , on peut atteindre dans G un sommet non couvert par T_u sans traverser un autre segment de S_u^1 . Donc, pour chaque segment $s_i \in S_u^1$ soit y_i un sommet non couvert par T_u qu'on peut atteindre dans G par un chemin p_i qui évite tous les autres segments de S_u^1 . Soit $P = \{p_1, p_2, \dots, p_r\}$ l'ensemble des chemins ainsi défini entre respectivement $y_1, y_2, \dots, y_r$. Vu que $\forall y_i, 1 \leq i \leq r, y_i \notin S_u^1$, alors chaque chemin p_i pour $1 \leq i \leq r$ est de longueur au moins 1. Donc en posant e_i la première arête de $p_i \in P$, on a pour $p_i, p_j \in P, i \neq j, e_i \neq e_j^1$. Étant donné que chaque segment de S_u^1 est adjacent à u dans T , alors en posant e'_i une arête reliant $s_i \in S_u^1$ à u pour tout $s_i \in S_u^1$, on a l'ensemble $P' = \{p'_1 = \{e'_1, e_1\}, p'_2 = \{e'_2, e_2\}, \dots, p'_r = \{e'_r, e_r\}\}$ qui a au moins r chemins de longueur 2 entre u et respectivement $y_1, y_2, \dots, y_r$.

Vu que $G \setminus T_u$ est connexe et $\{y_1, y_2, \dots, y_r\} \subseteq G \setminus T_u$, alors on construit r chemins disjoints de longueur 2 entre u et $G \setminus T_u$; ce qui est équivalent à la construction d'un $K_{2,r}$ dans G . On a donc une contradiction avec le fait que G est sans mineur $K_{2,r}$.

Donc $|S_u^1| < r$, ce qui termine la première partie.

Montrons maintenant par l'absurde que $|S_u^2| < r$.

En effet, supposons que $r \leq |S_u^2|$. D'abord, on va montrer par l'absurde que : pour tout segment $s \in S_u^2$, il existe un sommet $v_1 \in s$ et $v_2 \notin T_u$ tel que $\exists p_{v_1, v_2}$ un chemin entre v_1 et v_2 dans G tel que $p_{v_1, v_2} \cap S_u^2 = \emptyset$.

En effet, soit s_1 un segment de S_u^2 tel que tout chemin dans G reliant s_1 à un sommet $z \notin T_u$ utilise un autre segment s_2 de S_u^2 . Cela voudrait donc dire que tout chemin entre s_1 et z est couvert par T_u . A plus forte raison le chemin P entre s_1 et z , contenant s_2 et induit par le bord de la face extérieure du plongement planaire-exterieur de $G \setminus X$ est aussi couvert par T_u . Par conséquent, $s_1 \cup P \cup s_2$ constitue un seul segment de T_u car consécutifs sur le bord de la face extérieure de $G \setminus X$. Ce qui contredit l'hypothèse que les segments de T_u sont incompressibles.

D'où, à partir de s_1 on peut atteindre un sommet non couvert par T_u sans

¹Car, sinon les segments s_i, s_j contiendrait au moins un sommet en commun (une des extrémités de e_i); ce contredit l'hypothèse que les segments sont disjoints.

traverser un segment de S_u^2 . En appliquant donc le même procédé que pour S_u^1 , on aboutit à la même conclusion qui est que : $|S_u^2| < r$. Nous avons donc, montré que pour tout sommet $u \in X$, on a $|S_u| \leq |S_u^1| + |S_u^2| \leq 2r$. D'où on a $\sum_{u \in X} (S_u) \leq 2r|X| \leq 2kr$.

Calculons maintenant le nombre de segments de W_u . Nous allons montrer par l'absurde que $|W_u| < r$. En effet, supposons qu'il existe un sommet $u \in X$ tel que $|W_u| \geq r$. Alors, on peut former au moins $r - 1$ couples de segments consécutifs dans W_u tel que chaque couple induit au moins un segment qui est adjacent à un sommet de $X \setminus \{u\}$. En effet, posons $|W_u| = t$. L'ensemble W_u possède t segments ordonnés. Soit s_1, s_2 deux segments consécutifs de W_u tels que $\forall y_1 \in s_1, y_2 \in s_2$ on a $y_1 < y_2$. En d'autres termes, les sommets de s_1 sont numérotés avant les sommets de s_2 . Donc, il existe au moins un sommet $v \notin T_u$ tel que $\forall y_1 \in s_1, y_2 \in s_2$ on a $y_1 < v < y_2$. En d'autres termes, les sommets de s_1 sont numérotés avant v et le sommet v est numéroté avant les sommets de s_2 . En effet, si un tel sommet v n'existait pas, alors on aurait l'ensemble s_0 des sommets numérotés après les sommets de s_1 et avant les sommets de s_2 qui serait couvert par T_u . Dans ce cas, on aurait $s_1 \cup s_0 \cup s_2$ qui constituerait un seul segment de T_u ; ce qui contredirait le fait que les segments de T_u sont compactés au maximum.

Si on pose $W_u = \{s_1, s_2 \dots s_t\}$ tel que $\forall i, 1 \leq i \leq t$, s_i, s_{i+1} forment un couple de segments consécutifs. Alors, il existe $V_u = \{v_1, v_2, \dots, v_t\}$ un ensemble de t sommets tels que pour tout sommet $v_i \in V_u$, il existe deux segments $s_i, s_{i+1} \in W_u$ tels que $\forall y_i \in s_i$ et $\forall y_{i+1} \in s_{i+1}$ on a $y_i < v_i < y_{i+1}$. En effet, si un tel sommet v n'existait pas, alors $s_i \cup s_{i+1}$ constituerait un seul segment et donc contredirait le fait que les segments passant par u sont compactés au maximum.

De plus, vu que W_u est connexe et qu'aucun des segments de W_u n'est adjacent à u , alors on a W_u qui est connexe dans $G \setminus X$. De même, vu que T est connexe, $\forall v \in V$, il existe donc un chemin $p_{x,v}$ dans G entre x et v . Si $p_{x,v} \cap X = \emptyset$, alors $p_{x,v} \subseteq G \setminus X$.

Nous allons montrer qu'au plus un sommet v de V est tel que $p_{x,v} \cap (X \setminus \{u\}) = \emptyset$.

En effet, supposons qu'il existe deux sommets $v_i, v_{i+1} \in V$ tels que $p_{x,v_i} \cap (X \setminus \{u\}) = \emptyset$ et $p_{x,v_{i+1}} \cap (X \setminus \{u\}) = \emptyset$.

Soit $p_{v_i, s_{i+1}}$ le chemin entre v_i et s_{i+1} bordant la face extérieure de $G \setminus X$.

Soit $p_{s_{i+1}, v_{i+1}}$ le chemin entre s_{i+1} et v_{i+1} bordant la face extérieure de $G \setminus X$.

Soit $p_{s_i, x}$ le chemin entre s_i et x bordant la face extérieure de $G \setminus X$.

Ces chemins sont de longueur au moins 1 car tous ces sommets existent et sont tous différents. De plus, vu que ces chemins sont sur la face extérieure, alors chacun de ces chemins forme un segment. Ainsi, leur seules intersections possibles sont leurs extrémités. Alors les chemins $p_{x,v_i} \cup p_{v_i, s_{i+1}}$, $p_{x,v_{i+1}} \cup p_{s_{i+1}, v_{i+1}}$ et $p_{s_i, x} \cup p_{s_i, s_{i+1}}$ constituent 3 chemins disjoints de longueur au moins 2 entre x et s_{i+1} . On en déduit donc un mineur $K_{2,3}$; ce qui est absurde par hypothèse. Nous avons donc montré qu'au plus un sommet v de V est tel que $p_{x,v} \cap (X \setminus \{u\}) = \emptyset$. En d'autres termes, au moins $t - 1$ sommets de V sont adjacents à X dans T .

Donc, pour chaque couple s_i, s_{i+1} de segments de W_u , soit s'_i le segment induit adjacent à $X \setminus \{u\}$. Posons e_i l'arête reliant s'_i à $X \setminus \{u\}$. Vu que s_i, s_j, s'_i sont disjoints, alors il existe un chemin ρ_i sur le bord de la face extérieure de $G \setminus X$ qui

relie s_i et s'_i et ce chemin n'intersecte aucun segment de W . Le chemin $p_i = \rho_i \cup e_i$ est de longueur au moins deux et il relie u à $X \setminus \{u\}$. On peut construire un tel chemin pour tout couple de segments de W . De plus pour deux couples (s_i, s_{i+1}) et (s_j, s_{j+1}) pour $j \neq i$ on a deux chemins disjoints de longueur au moins deux.

Vu que $|W| = t$, alors on peut construire un ensemble P de t chemins disjoints de longueur au moins deux entre u et $X \setminus \{u\}$. Nous pouvons donc facilement en déduire un mineur $K_{2,t}$ et donc un mineur $K_{2,r}$; ce qui est une contradiction avec l'hypothèse que G est sans mineur $K_{2,r}$.

D'où, on a : $|W_u| \leq r$.

Donc, on a $\sum_{u \in X} |W_u| \leq r|X| \leq kr$.

Calculons maintenant le nombre de segments de G qui n'utilisent aucun sommet de X . En effet, on peut remarquer que si $x \notin X$, alors tout segment de G qui n'est descendant d'aucun sommet de X est accessible à x par de plus courts chemins dans $G \setminus X$.

Si $X = \emptyset$, alors G est planaire-extérieur et dans ce cas il est connu qu'il y a au plus un segment par arête. Dans ce cas le nombre de segments en x est au plus $|\Omega(x)|$.

Si X est non vide, alors calculons le cardinal de l'ensemble S_0 de segments qui n'utilisent aucun sommet de X . Considérons un sommet $v_i \in \Omega(x)$. Soit $S_{v_i} = T_{v_i} \cap S_0$ l'ensemble des segments de T_{v_i} qui n'utilisent aucun sommet de X . L'ensemble S_{v_i} contient au moins un segment, celui qui contient v_i .

Donc, on a $|S_0| \geq |\Omega(x)|$.

Soit s_1 un segment de S_{v_i} tel que $v_i \notin s_1$ et s_1 consécutif au segment contenant v_i . Le chemin dans T qui relie s_1 et v_i partage le plan du plongement de $G \setminus X$ en deux zones dont au moins une des deux zones contient un ensemble de sommets de $G \setminus X$ qui ne sont accessibles à x qu'en passant par un sommet de X . Cet ensemble de sommets induit donc au moins un segment adjacent à X dans T . Ainsi, si S_{v_i} contient k segments en plus du segment contenant v_i , en utilisant la relation d'ordre sur l'ensemble S_{v_i} on peut construire au moins k segments adjacents à X dans T . Si on pose k_i le nombre de segments de S_{v_i} supplémentaires au segment contenant v_i pour tout sommet $v_i \in \Omega(x)$, alors on a :

$$|S_0| \leq |\Omega(x)| + k_1 + k_2 + \dots + k_r.$$

Or $k_1 + k_2 + \dots + k_r$ est le nombre de segments induits par $\Omega(x)$ et adjacents à X dans T .

Donc, on a $k_1 + k_2 + \dots + k_r \leq \sum_{u \in X} |S_u| \leq 2kr$.

D'où, $|S_0| \leq |\Omega(x)| + 2kr$.

On a donc $\sum_{v \in \Omega} \text{seg}(T_v) \leq |\Omega(x)| + 5kr$.

Si $x \in X$, posons S l'ensemble des sommets qui utilisent $X \setminus \{x\}$ et soit $|S|$ le nombre de segments de S . Pour les mêmes raisons que dans le cas précédent, on a $|S| \leq 2kr$. Vu que seuls les sommets de S utilisent $X \setminus \{x\}$, alors $G \setminus \{S \cup X\} \cup \{x\}$ est connexe car sinon il existerait de sommets inaccessibles à x .

Posons H le sous-graphe de G induit par $G \setminus \{S \cup X_C\} \cup \{x\}$. Le sous-graphe $H \setminus \{x\}$ est planaire-extérieur et l'ensemble des sommets de H constituent au plus $2kr$ segments. En effet, chaque segment qui utilise X induit dans G un trou qui engendre deux segments disjoints consécutifs dans H . Vu que le nombre de trous possibles est égal au nombre de segments utilisant $X \setminus \{x\}$, alors le nombre de segments induits dans H est au plus $2kr$. Soit $V(x) = \{v_1, v_2, \dots, v_m\}$ l'ensemble des voisins de x dans H tel que $T_{v_1}, T_{v_2}, \dots, T_{v_m}$ couvre H .

Soit H_0 le graphe induit par H tel que H est un sous graphe induit de H_0 . En effet, H_0 contient H et entre tout couple de segments (s_1, s_2) de H , consécutifs sur le bord de la face extérieure du plongement de H_0 , on ajoute dans H_0 un chemin P_{s_1, s_2} de longueur égale au trou entre s_1 et s_2 . En d'autres termes si x_1 est le sommet de s_1 qui a le plus petit identifiant $Id(x_1)$ et x_2 le le sommet de s_2 qui a le plus grand identifiant $Id(x_2)$, alors $|P_{s_1, s_2}| = Id(x_2) - Id(x_1)$. Les sommets de P_{s_1, s_2} ont des numéros distincts et compris entre $Id(x_1)$ et $Id(x_2)$. Toutes les arêtes de P_{s_1, s_2} ont un poids infini ou suffisamment grand pour que ces arêtes ne soient sur aucun plus court chemin entre deux sommets de H .

Ensuite pour tout voisin v_i de x , si le poids dans H de l'arête (x, v_i) est p , alors tout arête adjacent à v_i dans H_0 , de poids p' est de poids $p + p'$. Il est clair que pour chaque sommet $v_i \in V(x)$, l'ensemble des sommets de H_0 qui sont plus proches de v_i contient l'ensemble des sommets de H qui utilisent v_i . Ainsi, si on regroupe pour chaque $v_i \in V(x)$ sous forme de segments, l'ensemble des sommets de H_0 qui sont plus proches de v_i , et en faisant la somme du nombre de segments sur tous les $v_i \in V(x)$, on obtient un nombre de segments qui majore le nombre de segment de H qui utilisent $V(x)$.

En effet, soit T_{v_1} un arbre de plus courts chemins couvrant H_0 enraciné en v_1 , un sommet arbitraire de $V(x)$. Il est connu par le fait que H_0 est planaire-extérieur, que tout sous-arbre de T_{v_1} contient un segment.

Pour chaque sommet $v_i \in V(x)$ soit T_{v_i} un arbre de plus courts chemins couvrant $H_0 \setminus (V(x) \setminus \{v_i\})$. Dans chaque arbre T_{v_i} , on conserve que les sommets qui ont une profondeur minimale dans T_{v_i} par rapport à tous les arbres T_{v_j} pour $j \neq i$ et $1 \leq i, j \leq m$.

Calculons le nombre de segments dans chacun de ces arbres. En effet, si on revient sur le processus de construction des arbres T_{v_i} pour $v_i \in V(x)$, on remarque que T_{v_1} est construit en premier et couvre tout le graphe dans un premier temps. En suite au fur et à mesure de la construction des autres arbres on enlève des sommets de T_{v_1} . Ce qu'on va montrer ici c'est que si on enlève un ensemble de sommets de T_{v_1} et que l'ensemble des sommets restant dans T_{v_1} forme plus d'un segment, alors on peut montrer l'existence d'un mineur $K_{2,3}$ dans H_0 . En effet soit s un segment tel que $T_{v_1} \setminus s$ forme plus d'un segment. Soit s_1, s_2 deux de ces segments tels que $\forall x \in s_1, y \in s, z \in s_2$, on $x \leq y \leq z$. Il existe un chemin p_1 dans H_0 entre

s_1 et s et $v_i \notin p_1$ de même p_2 entre s_2 et s car H_0 est bi-connexe. Les chemins $\rho_1 = T_{v_1}(v_1, s_1) \cup p_1$, $\rho_2 = T_{v_1}(v_1, s_2) \cup p_2$ et $\rho_3 = p_{(v_i, v_j)} \cup T_{v_j}(v_j, s)$ (avec $p_{(v_i, v_j)}$ un plus court chemin entre v_i et v_j dans H_0) induit un mineur $K_{2,3}$; ce qui contredit l'hypothèse que H_0 est planaire-extérieur. Donc T_{v_1} forme un segment.

Pour que l'arbre T_{v_1} soit un segment, alors les sommets enlevés de T_{v_1} sont sur les extrémités du segment de départ. Ainsi chacun des autres arbres a la possibilité de couvrir un segment dans chaque cotés. Ce qui veut donc dire que le nombre de segments par arbre autre que T_{v_1} peut être 2.

Montrons que ce nombre ne peut pas dépasser 2. En effet, supposons qu'il existe un arbre T_{v_j} $2 \leq j \leq m$ contenant plus de deux segments. Soit s_1, s_2, s_3 des segments disjoints et consécutifs de T_{v_j} . Il existe au moins un segment complémentaire s_{v_i} couvert par un arbre T_{v_i} avec $i \neq j$ et $2 \leq i \leq m$. Il existe aussi un segment s_{v_1} couvert par T_{v_1} et que pour tout sommet x_1 de s_{v_1} et tout sommet $x_j \in s_1, x'_j \in s_2, x''_j \in s_3$ on a soit $x_1 < \min\{x_j, x'_j, x''_j\}$ soit $x_1 > \max\{x_j, x'_j, x''_j\}$. Cela du fait que les segments couverts par T_{v_j} ne doivent pas augmenter le nombre de segments couverts par T_{v_1} . Nous allons montrer que dans ce cas on peut trouver un mineur $K_{2,3}$ dans H_0 .

En effet, si le plus court chemin p_1 entre v_1 et v_i ne contient pas v_j et le plus court chemin p_2 entre v_1 et v_j ne contient pas v_i , alors les chemins $\rho_1 = p_1 \cup T_{v_i}(v_i, s_{v_i})$, $\rho_2 = p_2 \cup T_{v_j}(v_j, s_{v_j})$ et $\rho_3 = T_{v_1}(v_1, s_{v_1}) \cup p(s_{v_1}, s_{v_j})$ (avec $p(s_{v_1}, s_{v_j})$ le chemin entre s_{v_1} et s_{v_j} bordant la face extérieure dans le plongement de H_0) induisent un mineur $K_{2,3}$ dans H_0 ; ce qui est une contradiction avec le fait que H_0 est planaire-extérieur.

Sinon, si le plus court chemin p_1 entre v_1 et v_j contient v_i , alors les chemins $\rho_1 = T_{v_j}(v_j, s_1) \cup p(s_1, s_2)$, $\rho_2 = T_{v_i}(v_i, s_{v_i}) \cup p(s_{v_i}, s_2)$ et $\rho_3 = T_{v_1}(v_1, s_{v_1}) \cup p(s_{v_1}, s_2)$ induisent un mineur $K_{2,3}$ dans H_0 ; ce qui est absurde par hypothèse.

Donc, le nombre de segment par arbre est au plus 2. Par conséquent, le nombre total de segments sur tous les arbres est au plus $2|V(x)|$. En enlevant les chemins qui étaient ajoutés à H pour former H_0 on obtient le nombre de segments de H en x . Le nombre de segments en x est donc au plus $2|V(x)| + 2kr \leq 2|\Omega(x)| + 2kr$.

Dans ce cas, le sommet x couvre tout au plus $2|\Omega(x)| + 2kr$ segments.

Donc, on a $\sum_{v \in \Omega} \text{seg}(T_v) \leq 2|\Omega(x)| + 4kr \leq 2|\Omega(x)| + 5kr$.

Ce qui termine la preuve. $\square$

4.3.1.2 Partition de G

Nous venons de donner une généralisation du théorème 4.11. Il reste donc à donner une généralisation du lemme 4.6 avant de passer à l'élaboration du schéma de routage. On peut remarquer que dans le cas où le graphe est valué, le lemme 4.6 n'est plus vérifié. Il faut donc trouver la partition du graphe permettant d'avoir une propriété similaire à celle du lemme 4.6. C'est dans ce cadre que nous présentons la partition ci-dessous sous le nom de *RSU-partition*.

Définition 4.13 (RSU-partition) Soient G un graphe et $\mathcal{T}$ un arbre couvrant G de plus courts chemins, enraciné en un sommet arbitraire.

Pour tout sommet $x \in G$, une RSU-partition de G en x suivant $\mathcal{T}$ est une partition des sommets de G en trois parties :

- (i) $\mathbb{R}(x)$ ($\mathbb{R}$ comme related) l'ensemble des descendants de x dans $\mathcal{T}$, noté $\mathcal{T}_x$.
Notons que x est dans $\mathbb{R}(x)$.
- (ii) $\mathbb{S}(x)$ l'ensemble des sommets y de $G \setminus \mathbb{R}(x)$, ayant au moins un ancêtre z dans $\mathcal{T}$ voisin de x .
- (iii) $\mathbb{U}(x)$ l'ensemble des sommets y de G tels que $y \notin \mathbb{R}(x)$ et $y \notin \mathbb{S}(x)$.

On dit qu'un sommet x est un *proche* d'un sommet y si $\exists z \in N(x)$ et $y \in \mathbb{R}(z)$. Deux sommets x et y sont proches si $y \notin \mathbb{R}(x)$ et $y \in N(x)$.

Le lemme suivant est similaire au lemme 4.6 :

Lemme 4.14 Soient G un graphe (k, r) -cellulaire et $\mathcal{T}$ un arbre couvrant G de plus courts chemins et soit une RSU-partition de G en un sommet x de G suivant $\mathcal{T}$. Alors pour tout arbre $\mathcal{T}$ couvrant G de plus courts chemins enraciné en x , il y a au plus $r - 1$ voisins $v_1, v_2, \dots, v_{r-1}$ de x tels que $\mathbb{U}(x) \subseteq \{\mathcal{T}_{v_1} \cup \mathcal{T}_{v_2} \cup \dots \cup \mathcal{T}_{v_{r-1}}\}$.

Démonstration : Soit x un sommet de G et $\mathbb{U}(x)$ l'ensemble des sommets de G qui ne sont ni dans $\mathbb{S}(x)$ ni dans $\mathbb{R}(x)$. Soit $\mathcal{T}$ un arbre couvrant G de plus courts chemins, enraciné en x .

Soit $V_x = \{v_1, v_2, \dots, v_t\}$ un ensemble de voisins de x tels que $\mathbb{U}(x) \subseteq \{\mathcal{T}_{v_1} \cup \mathcal{T}_{v_2} \cup \dots \cup \mathcal{T}_{v_{r-1}}\}$.

- (i) Si $t \leq r - 1$, alors on a $|V_x| \leq r$ et le lemme est vrai.
- (ii) Supposons que $r \leq t$ et montrons que dans ce cas, G contient un mineur $K_{2,r}$.

En effet, $\forall v_i \in V_x$, il existe au moins un sommet $y_i \in \mathbb{U}(x)$ tel que (x, v_i) est la première arête d'un plus court chemin entre x et y_i . Pour chaque sommet $v_i \in V_x$, soit y_i un sommet de $\mathbb{U}(x) \cap \mathcal{T}_{v_i}$ et posons p_i le chemin dans $\mathcal{T}_{v_i}$ reliant x à y_i . Posons $P = \{p_1, p_2, \dots, p_t\}$.

On peut facilement remarquer qu'aucun sommet $v_i \in V_x$ n'est ancêtre d'un sommet de $\mathbb{U}(x)$ dans $\mathcal{T}$. En effet si un sommet $y_i \in \mathbb{U}(x)$ était descendant d'un sommet $v_i \in V_x$ dans $\mathcal{T}$ alors on aurait x qui serait voisin d'un ancêtre de y_i et dans ce cas y_i serait dans $\mathbb{S}(x)$ ce qui est une contradiction avec le fait que $y_i \notin \mathbb{S}(x)$ pour tout $y_i \in \mathbb{U}(x)$.

Par conséquent, pour chaque chemin $p_i \in P$, il existe un sommet $v'_i \neq v_i$ et $(v_i, v'_i) \in p_i$. Une illustration est donnée sur la figure 4.1. Nous avons ainsi t chemins distincts de longueur 2 reliant x à chaque sommet v'_i avec $1 \leq i \leq t$. Ce qui voudrait aussi dire que tous les chemins dans P sont au moins de longueur 2.

Posons $\mathcal{T}_0$ l'union des chemins dans $\mathcal{T}$ reliant la racine r de $\mathcal{T}$ à chacun des sommets y_i .

On peut facilement remarquer que $\mathcal{T}_0$ est un arbre et qu'il ne contient aucun sommet $v_i \in V_x$. En effet si un sommet v_i était dans $\mathcal{T}_0$, alors c'est soit parce que v_i lui-même est dans $\mathbb{U}(x)$; ce qui est impossible d'après notre hypothèse, soit c'est parce que v_i est un ancêtre d'un sommet y_i de $\mathbb{U}(x)$ dans $\mathcal{T}$ ce qui est aussi absurde car x n'est voisin d'aucun ancêtre dans $\mathcal{T}$ d'un sommet de l'ensemble $\mathbb{U}(x)$ par hypothèse.

Vu que $\mathcal{T}_0$ est connexe car étant un arbre, en contractant $\mathcal{T}_0$ en un sommet t_0 , l'ensemble P devient un ensemble à t chemins de longueurs au moins 2 entre x et t_0 ; ce qui constitue un mineur $K_{2,t}$ dans G .

Comme on a supposé en hypothèse que $r \leq t$, alors on a G qui contient un mineur $K_{2,r}$; ce qui est une contradiction au fait que G est sans mineur $K_{2,r}$. On a donc $t < r$ et le lemme 4.14 est vrai. $\square$

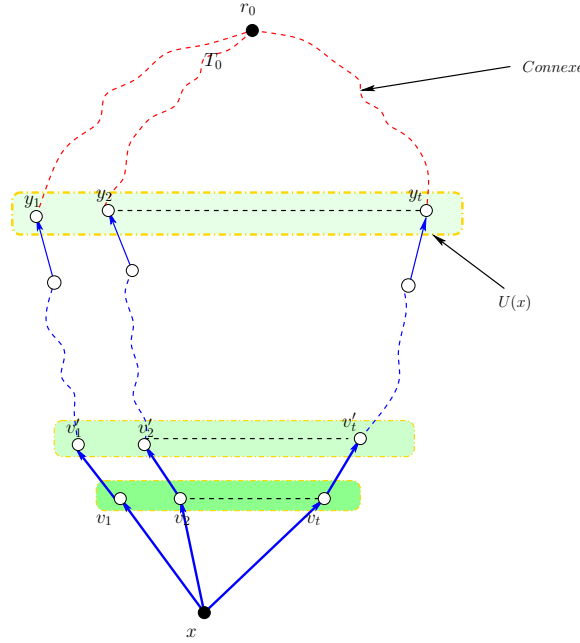


FIG. 4.1: Construction d'un mineur $K_{2,r}$ de G .

4.3.1.3 Construction d'un schéma de routage de plus courts chemins pour les graphes (k, r) -cellulaires

Soit G un graphe (k, r) -cellulaire d'ordre n . Nous avons besoin de numérotter les sommets de G pour ensuite exploiter les propriétés liées à cette numérotation dans la mise en oeuvre du schéma de routage.

Fonctions de numérotation des sommets de G : Le graphe G étant (k, r) -cellulaire, alors grâce au lemme 4.12 chaque composante bi-connexe C de G admet une numérotation qui lui permet d'avoir une f -propriété d'intervalle. Nous avons besoin de numérotter chaque composante bi-connexe pour mettre en place la numérotation permettant d'avoir cette f -propriété d'intervalle. Cette numérotation se fait de la manière suivante :

Soit C_0 une composante bi-connexe de G et r_0 un sommet de C_0 . Soit $\mathcal{T}$ un arbre enraciné construit à partir de G en remplaçant chaque composante bi-connexe de G par une étoile dont les extrémités représentent les sommets d'articulations de la composante et le centre représente la composante. La racine de $\mathcal{T}$ est le sommet représentant la composante C_0 . Soit une numérotation des composantes bi-connexes suivant un DFS sur $\mathcal{T}$ par des entiers entre 1 et $|\mathcal{T}|$. Cette numérotation sera utile à la fonction suivante numérotant les sommets.

Soit Id_2 la fonction de numérotation des sommets de G qui commence par numérotter les sommets appartenant à la composante bi-connexe C_0 en suite les sommets appartenant à la deuxième composante bi-connexe de plus petit identifiant et ainsi de suite.

Pour chaque composante bi-connexe C , on numérote en premier l'ensemble X_C des sommets tels que $C \setminus X_C$ est planaire extérieur. En suite en considérant un plongement planaire-extérieur du sous-graphe induit par les sommets de $C \setminus X_C$, on numérote les sommets de $C \setminus X_C$ suivant un parcours bordant la face extérieure. Les sommets appartenant à une même composante bi-connexe sont numérotés par des entiers consécutifs.

Les sommets de C_0 ont des numéros compris entre 1 et $|C_0|$. Ensuite on procède à un décalage qui consiste à choisir un sommet d'articulation a_0 de C_0 qui est tel que $Id_2(a_0)$ est minimal par rapport à tous les autres sommets d'articulation de C_0 , calculer le nombre de sommets d des composantes connexes de $C_0 \setminus a_0$ et pour tout sommet $x \in C_0$ tel que $Id_2(x) > Id_2(a_0)$, faire $Id_2(x) = Id_2(x) + d$.

Recommencer ce décalage en choisissant le deuxième sommet d'articulation a_1 qui est tel que $Id_2(a_1)$ est minimal. Une fois ce décalage effectué sur tous les sommets d'articulation de C_0 , on recommence récursivement la procédure avec comme composante bi-connexe la composante C_1 ayant le plus petit identifiant après celui de C_0 . Les sommets de C_1 sont numérotés par des entiers compris entre $|C_0| + 1$ et $|C_0| + |C_1|$ et un décalage est fait sur tous les sommets d'articulations de C_1 sauf celui qui le relie à C_0 .

De manière générale, si C_i est la i^{eme} composante bi-connexe choisie, alors la numérotation attribuée aux sommets de C_i des entiers compris entre $\sum_{k=1}^{i-1} |C_k| + 1$ et $\sum_{k=1}^{i-1} |C_k| + |C_i|$, puis effectuée un décalage suivant les sommets d'articulation de $C_i \setminus \{C_i \cap C_{i-1}\}$. Pour tout sommet x , $Id_2(x)$ est le numéro de x issu de la numérotation Id_2 .

Cette numérotation Id_2 fait qu'on peut avoir la f -propriété d'intervalle dans chaque composante bi-connexe. Cette numérotation sera utile pour la suite. Nous avons aussi besoin de la fonction de numérotation Id_1 suivante pour l'implémentation de notre schéma de routage.

Soit T un arbre couvrant G de plus courts chemins, enraciné en r_0 .

Soit Id_1 une fonction de numérotation des sommets de G suivant un DFS sur T . Pour tout sommet x , $Id_1(x)$ est le numéro de x issu de la numérotation Id_1 .

Construction du schéma de routage : Pour tout sommet x de G :

1. Soit $(\mathbb{R}(x), \mathbb{S}(x), \mathbb{U}(x))$ une RSU-partition de G en x basée sur l'arbre T .
2. Soit $TABLE\text{-}\mathbb{R}(x)$ une table de routage construite à partir de l'arbre T dans laquelle seuls les descendant de x dans T sont préservés. Grâce à la fonction de numérotation Id_1 , l'ensemble des descendants de x dans T constitue un intervalle d'entiers modulo n .
3. Soit $TABLE\text{-}\mathbb{S}(x)$ une table de routage construite à partir de la RSU-partition de G en x dans laquelle seuls les sommets de $\mathbb{S}(x)$ qui utilisent des ports $\leq 2r$ sont préservés. Ainsi pour toute destination $y \in \mathbb{S}(x)$, si y utilise un port p tel que $p \leq 2r$, alors le p^{eme} champ de $TABLE\text{-}\mathbb{S}(x)$ contient un segment I

tel que $Id_1(y) \in I$. Toute destination $z \in \mathbb{S}(x)$ qui utilise un port p tel que $p > 2r$, stocke dans un champ spécial P de son adresse le port par lequel on doit passer pour aller de x vers z par un plus court chemin. Nous donnerons des explications sur comment il faut numéroter les ports.

4. Soit $TABLE-\mathbb{U}(x)$ une table de routage construite à partir de la RSU-partition de G en x dans laquelle seuls les sommets de $\mathbb{U}(x)$ sont préservés. En effet, l'ensemble des destinations passant par le même port est compactés au maximum en segments suivant les identifiant Id_2 de telle sorte que pour toute destination $y \in \mathbb{U}(x)$, si y utilise le port p , alors le p^{eme} champ de $TABLE-\mathbb{U}(x)$ contient un segment I tel que $Id_2(y) \in I$.
5. Soit $ADRESSE-(x)$ l'adresse du sommet x constituée par une structure de donnée contenant :
 - a) l'identifiant $Id_1(x)$ de x issu de la fonction de numérotation Id_1 .
 - b) l'identifiant $Id_2(x)$ de x issu de la fonction de numérotation Id_2 .
 - c) un champ spécial P contenant tout port p traversé pour aller en x depuis tout sommet $x_0 \in G$ tel que $x \in \mathbb{S}(x_0)$ et $p > 2r$. Le port p sera stocké par x dans un emplacement réservé i_{x_0} de P pour le sommet x_0 . Ainsi, pour tout message destiné à x en x_0 , le sommet x_0 pourra facilement vérifier que $x \in \mathbb{S}(x_0)$ et $x \notin TABLE-\mathbb{S}(x)$ et le cas échéant récupérer le port de sortie donné par $ADRESSE-(x)[i_{x_0}]$; ce port est stocké dans l'emplacement i_{x_0} du champ P de $ADRESSE-(x)$.

L'algorithme de routage : L'algorithme de routage présenté ci-dessous est élémentaire. Son originalité se trouve dans son implémentation.

ROUTE : de $x \in G$ vers $y \in G$:

1. si $y \in TABLE-\mathbb{R}(x)$, retourner $TABLE-\mathbb{R}(x)[y]$, sinon
2. si $y \in TABLE-\mathbb{U}(x)$, retourner $TABLE-\mathbb{U}(x)[y]$, sinon
3. si $y \in TABLE-\mathbb{S}(x)$ retourner $TABLE-\mathbb{S}(x)[y]$, sinon
- 4 retourner $ADRESSE-(y)[i_x]$

4.3.1.4 Explications de l'algorithme

- (A) Vu que pour tout sommet x , les ensembles $\mathbb{R}(x)$, $\mathbb{S}(x)$ et $\mathbb{U}(x)$ sont issus d'une RSU-partition, alors tout sommet $y \in G$ est soit $\mathbb{R}(x)$, soit dans $\mathbb{S}(x)$ ou dans $\mathbb{U}(x)$. Par ailleurs une numérotation des sommets suivant un parcours en profondeur de l'arbre T permet d'avoir un segment d'entier pour chaque sous-arbre de T . Du fait que chaque sommet connaît le segment représentant son sous-arbre, il est facile de tester si $y \in \mathbb{R}(x)$ et de retourner le numéro de port par lequel x doit faire transiter le message. C'est une opération en temps constant.
- (B) Pour tout sommet x , tester si un sommet y est dans $\mathbb{U}(x)$ est simple. Il suffit de chercher un segment contenant y dans la table $TABLE-\mathbb{U}(x)$. En effet, cette table comme nous l'avons expliqué plus haut, est telle que chacune de ses entrées contient non seulement le port associé mais aussi l'ensemble des segments qui

l'utilisent. Ainsi, si y est dans un de ses segments, alors y utilisent le port p associé à cette entrée. Si la taille de la tables est t , alors en supposant qu'elle soit préalablement triée nous pouvons faire une recherche dichotomique et trouver y en temps $O(\log t)$. Nous allons déterminer la valeur exacte de t lorsque nous détaillerons l'implémentation de la table.

- (C) Si le sommet y n'est ni dans $\mathbb{R}(x)$ ni dans $\mathbb{U}(x)$, alors y est dans $\mathbb{S}(x)$. Dans ce cas il y a deux possibilités : soit c'est x qui détient l'information, soit l'information est dans l'adresse de y .
 - (a) Si l'information est détenue par x alors c'est que y utilise un des $2r$ premiers ports desservant $\mathbb{S}(x)$. Dans ce cas le port de sortie est donné par $TABLE\text{-}\mathbb{S}(x)[y]$. Cela consiste à chercher dans la table $TABLE\text{-}\mathbb{S}(x)$ un segment I tel que $y \in I$. Nous allons montrer dans la partie implémentation, que cette table contient au plus $O(f(r))$ segments. Donc, une recherche dichotomique d'un segment dans cette table se fait en $O(\log f(r))$; en supposant que le tableau soit préalablement trié.
 - (b) Si le test est négatif, alors c'est y qui détient l'information dans son adresse. Il reste au sommet x d'aller chercher l'information dans le champ P de l'adresse de y . Vu que chaque sommet connaît l'emplacement qui lui est réservé dans le champ P , alors cette opération consistera à aller lire directement dans un tableau. C'est une opération en temps constant.

L'algorithme s'exécute donc en $O(\log f(r)) + O(\log t)$.

4.3.1.5 Implémentation des tables de routage

1. $TABLE\text{-}\mathbb{R}(x)$ est une table de routage de x qui stocke l'ensemble des informations pour router vers les destinations dans $\mathbb{R}(x)$. En effet, si on considère la numérotation Id_2 , alors, pour chaque sommet x l'ensemble des descendants $\mathbb{R}(x)$ de x dans T forme un intervalle d'entiers. Chaque sommet x stocke dans $TABLE\text{-}\mathbb{R}(x)$ l'intervalle l'entier correspondant à $\mathbb{R}(x)$. Pour toute destination $y \in \mathbb{R}(x)$, si y appartient au sous-arbre de x le plus lourd, alors x stock l'arête ou le numéro de port qui permet de router vers y dans $TABLE\text{-}\mathbb{R}(x)$ de telle sorte que $TABLE\text{-}\mathbb{R}(x)[y]$ donne l'arête qui mène vers y . La table $TABLE\text{-}\mathbb{R}(x)$ contient donc, le segment compactant l'ensemble des destinations du sous-arbre le plus lourd et le numéro de port associé. La table de routage $TABLE\text{-}\mathbb{R}(x)$ est donc de taille $O(\log n)$ bits.

Si y est dans un des sous-arbres les moins lourds, alors y stocke l'arête ou le numéro de port dans son adresse de telle sorte que le sommet x puisse accéder facilement à cette information.

Ainsi, pour tout sommet y $TABLE\text{-}\mathbb{R}(x)[y]$ retrouve l'arête ou le numéro de port pour aller de x en y , en utilisant uniquement les informations stockées dans $TABLE\text{-}\mathbb{R}(x)$ et dans l'adresse de y . En utilisant la technique d'implémentation de [FG01b, TZ01] on peut implémenter les informations à stocker dans l'adresse de x en $O(\log n)$ bits.

2. La table $TABLE\text{-}\mathbb{S}(x)$ est une table de routage permettant de router vers les destinations de $\mathbb{S}(x)$. Dans cette partie, nous allons expliquer comment

implémenter cette table. Pour cela, nous avons besoin que chaque sommet x puisse identifier de manière exacte la composante bi-connexe $C(x)$ à laquelle il appartient. Ainsi, pour chaque sommet x , la composante bi-connexe $C(x)$ est celle qui contient x si x n'est pas un sommet d'articulation et, dans le cas où x est un sommet d'articulation $C(x)$ est la composante bi-connexe contenant l'arc qui relie x à son père dans T . Il y en a au plus une, vu que tout arc de G est dans au plus une composante bi-connexe.

Supposons que chaque sommet x admet une étiquette spéciale qui n'est rien d'autre qu'un booléen qui est *VRAI* si $\mathbb{S}(x) \neq \emptyset$ et *FAUX* sinon. Dans la suite, nous appellerons sommet spécial tout sommet ayant une étiquette spéciale à *VRAI*. Les sommets spéciaux sont les seuls sommets qui ont besoin d'implémenter la table $TABLE\text{-}\mathbb{S}(x)$. Pour cela, pour tout sommet spécial x et tout sommet $v \in N(x)$, nous allons compacter sous forme de segments l'ensemble des destinations de $\mathbb{S}(x)$ passant par v . Ainsi, l'ensemble des segments obtenus dans tous le voisinage de x permettra la construction de la table de routage $TABLE\text{-}\mathbb{S}(x)$.

Ainsi, pour chaque sommet spécial x , soit T'_x un arbre couvrant $\mathbb{S}(x)$ de plus courts chemins dans G , enraciné en x . Donc, pour tout voisin v de x , l'ensemble des sommets de T'_v sont les destinations de $\mathbb{S}(x)$ qui utilisent v . Or par définition, tout sommet de $\mathbb{S}(x)$ qui utilisent v est un descendant de v dans T . De plus, nous savons que T_v est un segment suivant les identifiants Id_1 . Nous allons partager la table $TABLE\text{-}\mathbb{S}(x)$ en deux parties. La première partie contient pour tout sommet $v \in N(x)$, sous forme de segments d'identifiants Id_2 l'ensemble des destinations de $C(x) \cap \mathbb{S}(x)$ passant par v . La deuxième partie contient pour tout sommet $v \in N(x)$ toutes les destinations $y \notin C(x)$ passant par v tel que $y \in \mathbb{S}(x)$; sous forme de segments suivant la numérotation Id_1 .

De plus, pour chaque sommet spécial x l'ensemble $N \subseteq N(x)$ des voisins v de x tels qu'il existe au moins une destination y de $\mathbb{S}(x)$ passant par v est partagé en deux groupes N_1 et N_2 . L'ensemble N_1 est toujours maximisé dans N . Si $|N| \geq 2r$, alors N_1 contient au moins $2r$ sommets et l'ensemble $N_2 = N \setminus N_1$. Sinon, on a $N_1 = N$ et $N_2 = \emptyset$.

Nous demandons à chaque destination passant par N_2 de stocker le numéro de port de sortie en x . Le sommet x stockera alors dans la table $TABLE\text{-}\mathbb{S}(x)$, sous forme de segments comme nous venons de le décrire, les destinations passant par N_1 .

Pour expliquer comment ce partitionnement va se faire, nous avons besoin des quelques spécifications suivantes :

Pour tout sommet y , soit X_y l'ensemble des sommets tels que $\forall x \in X_y, y \in \mathbb{S}(x)$.

Soit $\mathcal{T}$ l'arborescence construite de la manière suivante :

Soit y un sommet tel que $X_y \neq \emptyset$.

- (i) Soit P_y un sac vide. Faire $P_y = X_y \cup y$.
- (ii) Pour tout sommet y' tel que $X_{y'}$ intersecte P_y , faire $P_y = P_y \cup X_{y'} \cup y'$.

- (iii) Faire $\mathcal{T} = \mathcal{T} \cup P_y$ et choisir w un sommet tel que $w \notin \mathcal{T}$ tel que $X_w \neq \emptyset$ et recommencer en (i) si w existe en créant un nouveau sac P_w .

Dans chaque sac de $\mathcal{T}$, nous introduisons une relation d'ordre entre les sommets spéciaux. En effet, pour tout couple de sommets spéciaux, le plus petit est celui qui est plus proche de la racine de T .

Le procédé que nous allons décrire ici, traitera les sommets spéciaux un à un du plus petit au plus grand. Si un ensemble A d'éléments sont tous des minimas, alors le minimum sera choisi arbitrairement parmi les éléments de A . Ce cas se produit si un ensemble $A = \{x_1, \dots, x_k\}$ de sommets spéciaux sont adjacents à la fois à un même sommet z et $\forall x_i \in A, z \in \mathbb{S}(x_i)$. On dit dans ce cas que A est déterminé par le sommet z . Le rang de A est déterminé par l'identifiant de z issu de la fonction de numérotation Id_1 . Il peut arriver qu'un sommet spécial x appartienne à la fois à plusieurs ensembles de minimas. Dans ce cas, le rang de x est déterminé par son rang dans l'ensemble qui à le rang le plus petit parmi tous les ensembles contenant x .

Dans chaque sac de $\mathcal{T}$, on prend un minimum x . Soit $V(x)$ l'ensemble des voisins de x tels que $\forall y \in V(x), T'_y \cap \mathbb{S}(x) \neq \emptyset$. Par exemple sur la figure 4.2, le sommet y_1 est le premier minimum choisi et $V(y_1) = \{v_1, v_2, v_3\}$.

Pour tout $y \in V(x)$ soit C_y la composante connexe induite par l'union de T'_y avec l'ensemble X des sommets spéciaux tels que $\forall z \in X, \mathbb{S}(z) \cap T'_y \neq \emptyset$. Sur la figure 4.2, on a T'_{v_1} qui induit la composante connexe C_{v_1} .

On remarque que l'intersection entre deux composantes connexes C_y, C_z est un ensemble de sommets spéciaux. En effet, les sommets de T'_y sont des descendants de y dans T et les sommets de T'_z sont des descendants de z dans T . S'il existait un sommet t tel que $t \in T'_y \cap T'_z$, alors il existerait un cycle dans T ; ce qui est absurde car T est un arbre. Sur la figure 4.2, on remarque que $C_{v_1} \cap C_{v_2} \neq \emptyset$ et est composée que par des sommets spéciaux.

Par ailleurs, pour tout ensemble $W \subseteq V(x)$ minimal tel que $G \setminus \{W \cup \{x\}\}$ engendre une composante connexe C , on a $|W| < r$. En effet, si un ensemble $W \subseteq V(x)$ minimal est tel que $r \leq |W|$ et que $G \setminus \{W \cup \{x\}\}$ engendre une composante connexe C , alors G admet un mineur $K_{2,r}$. En effet, chaque sommet de W est adjacent à la fois à x et au moins un sommet de C ; W est donc un séparateur minimal de G . Ce qui veut donc dire que G admet un mineur $K_{2,r}$; ce qui est une contradiction avec l'hypothèse que G est sans mineur $K_{2,r}$. Par conséquent, tout ensemble minimal W de sommets de $V(x)$ tel que $G \setminus \{W \cup \{x\}\}$ engendre une composante connexe contient moins de r sommets. Donc, toute composante connexe C de $G \setminus \{W \cup \{x\}\}$ est couverte par au plus r sous-arbres de T'_x . En effet, tout chemin entre x un sommet y de C utilise forcément W .

Nous pouvons à présent procéder à l'implémentation de la table de routage du minimum spécial x choisi. Pour cela, nous avons besoin de numéroter les ports de sortie en x .

On numérote les ports suivant les poids des sous arbres qu'ils desservent. Ainsi le sous-arbre le plus lourd a le port 1 et le k^{eme} sous-arbre le plus lourd a le

port numéro k . Cette numérotation de port s'applique à tous les sommets spéciaux de G .

Une fois les ports numérotés, le sommet x stocke dans sa table de routage les $2r$ premiers ports. On sait que la composante $C(x)$ a la f -propriété d'intervalle. Donc l'ensemble des destinations de $C(x)$ qui utilisent les $2r$ premiers ports forme au plus $O(f(r))$ segments.

Par ailleurs, à chacun de ces $2r$ ports correspond une arête (x, v) telle que $v \in N(x)$ et $v \in \mathbb{S}(x)$. L'ensemble des sommets de $\mathbb{S}(x)$ qui utilisent v sont des sommets de T_v . De plus tout sommet qui n'est pas dans $C(x)$ utilise forcément un sommet d'attachement de $C(x)$. Or l'ensemble des sommets qui utilisent un attachement a de $C(x)$ sont les sommets de T_a ; donc constitue un segment. Donc en regroupant pour chaque port p l'ensemble des destinations passant par p s'il y en a au moins un attachement, on stocke au plus $2r$ segments de plus. En effet, l'ensemble des destination passant par le même port est un segment suivant la numérotation suivant l'arbre T .

La table $TABLE\text{-}\mathbb{S}(x)$ se compose donc en deux parties. La première partie contient sous forme de segments l'ensemble des destinations de $C(x)$ qui utilisent les $2r$ premiers ports. La deuxième partie contient l'ensemble des destinations de G qui utilisent les $2r$ premiers ports utilisés par au moins un attachement. On remarque que la deuxième partie peut contenir des informations déjà présentes dans la première partie. La recherche d'un port se fera donc en deux étapes, d'abord sur la première partie et ensuite sur la deuxième partie. En supposant que chaque partie soit triée, la recherche se fera en $O(\log f(r))$.

Toute destination $y \in \mathbb{S}(x)$ qui n'est pas dans les $2r$ premiers sous-arbres les plus lourds, stocke dans un champ spécial P de son adresse le port permettant de router de x vers y . Nous allons en fait montrer dans la construction des adresses que chaque sommets y stocke dans son adresse au plus $\log n$ numéros de ports. Nous demandons ainsi à chaque sommet spécial x de choisir un entier entre 1 et $\lceil \log r \rceil$. En effet, si x choisi l'entier k , alors toute destination y de $\mathbb{S}(x)$ va stocker le port permettant de router de x vers y à la k^{eme} position du champs P . Ainsi, du fait que x connaît cet entier, alors il pourra trouver en temps constant le numéro de ports en ayant juste l'adresse de y .

Ainsi termine la construction de la table $TABLE\text{-}\mathbb{S}(x)$. Cette table est de taille au plus $O(r \log n)$ bits.

En appliquant récursivement ce procédé sur chaque composante connexe de $G \setminus \{W \cup \{x\}\}$ on construit pour chaque sommet spécial x une table de routage de $O(r \log n)$ bits pour les destinations de $\mathbb{S}(x)$. Par exemple sur la figure 4.2, le second minimum considéré est le sommet y_2 . Pour ce sommet les nouvelles composantes connexes associées à chacun des ports utilisés par des sommets de $\mathbb{S}(x)$ sont représentées par des patates rouges.

3. $TABLE\text{-}\mathbb{U}(x)$ est une table de routage de x qui stocke l'ensemble des informations pour router vers les destinations dans $\mathbb{U}(x)$. En effet, chaque entrée de la table $TABLE\text{-}\mathbb{U}(x)$ est associé à une arête ou un port de sortie en x . Ainsi, l'entrée associée à un port p contient le numéro de port plus l'ensemble des

segments qui utilisent ce port. De ce fait, si l'entrée i de la table contient un segment s tel que $y \in s$, alors y utilise le port associé à cette entrée.

D'après le lemme 4.14, le nombre de voisins de x qui couvrent $\mathbb{U}(x) \cap C(x)$ est au plus $r - 1$.

Si x n'est pas un séparateur de G , alors tout voisin de x est dans $C(x)$. Donc le nombre de voisins de x qui couvrent $\mathbb{U}(x)$ est dans $C(x)$ et donc ne peut pas dépasser $r - 1$. Donc au plus il y a $r - 1$ numéros de ports à stocker par x , c'est-à-dire $O(r \log n)$ bits. De plus $C(x)$ ayant une f -propriété d'intervalle, alors le nombre de segments de $C(x) \cap \mathbb{U}(x)$ est au plus $O(f(r))$.

Par ailleurs tout sommet $y \in G \setminus C(x)$ est dans une composante connexe attachée à $C(x)$ par un sommet séparateur a . Nous allons montrer que la f -propriété d'intervalle de $C(x)$ peut être étendue en x à G . En effet, soit $s = \{x_1, x_2, \dots, a, x_j, \dots, x_k\}$ le segment de $C(x) \cap \mathbb{U}(x)$ contenant a , dans la f -propriété d'intervalle. Soit $C_1, \dots, C_l$ les composantes connexes ne contenant pas x et engendrées par $G \setminus a$. Posons $d = |C_1| + \dots + |C_l|$. Par définition de la fonction de numérotation Id_2 on a $\forall j \leq i \leq k, Id_2(x_i) = Id_2(a) + d + i - j + 1$. De plus cette fonction de numérotation attribue à tout sommet de $C_1 \cup \dots \cup C_l$ un identifiant compris entre $a + 1$ et d . Donc, suivant les identifiants issus de Id_2 , $s \cup C_1 \cup \dots \cup C_l$ constitue un segment modulo n . Nous avons donc étendu la f -propriété d'intervalle à G en x . Le nombre d'intervalles en x est donc d'au plus $O(f(r))$.

Si x est un séparateur, $G \setminus \{x\}$ est non connexe et tout sommet y qui n'est pas dans la même composante connexe que r est dans T_x . Vu que $T_x = \mathbb{R}(x)$, alors $y \notin \mathbb{U}(x)$.

Par ailleurs, on sait que $C(x)$ a la f -propriétés d'intervalle. Soit $a \neq x$ un séparateur de $C(x)$. Soit $s = \{x_1, x_2, \dots, a, x_j, \dots, x_k\}$ le segment de $C(x)$ contenant a dans la f -propriété d'intervalle. Soient $C_1, \dots, C_l$ les composantes connexes ne contenant pas x et engendrées par $G \setminus a$. Posons $d = |C_1| + \dots + |C_l|$. Par définition de la fonction de numérotation Id_2 on a $\forall j \leq i \leq k, Id_2(x_i) = Id_2(a) + d + i - j + 1$. De plus cette fonction de numérotation attribue à tout sommet de $C_1 \cup \dots \cup C_l$ un identifiant compris entre $a + 1$ et d . Donc, suivant les identifiants issus de Id_2 , $s \cup C_1 \cup \dots \cup C_l$ constitue un segment modulo n . Nous avons donc étendu la f -propriété d'intervalle à G en x . Le nombre d'intervalles en x est donc d'au plus $O(f(r))$.

D'où la table $TABLE\text{-}\mathbb{U}(x)$ est de taille au plus $O(r \log n) + O(f(r) \log n)$ bits.

La table de routage de chaque sommet x est donc constituée par les trois tables de routages précédentes à savoir $TABLE\text{-}\mathbb{R}(x)$, $TABLE\text{-}\mathbb{S}(x)$ et $TABLE\text{-}\mathbb{U}(x)$. Donc, la table de routage pour chaque sommet est en $O(f(r) \log n)$ bits.

4.3.1.6 Implémentation des adresses

Pour tout sommet $x \in G$ son adresse notée $ADRESSE(x)$, est une structure de donnée contenant :

1. l'identité de x issue de la numérotation Id_1 .

2. l'identité de x issue la numérotation Id_2 .
3. Un champ spécial P de $ADRESSE(x)$ contenant l'ensemble des numéros des ports permettant d'aller en x à partir de tout sommet $y \in G$ tel que $x \in \mathbb{S}(y)$ et x n'est pas couvert par les $2r$ sous-arbres de T'_y de poids fort.

On remarque que les deux premiers champs de $ADRESSE(x)$ sont chacun de taille $O(\log n)$ bits. Il reste donc à quantifier la taille du troisième champ P de $ADRESSE(x)$. En effet on sait que la racine r de T est telle que $\mathbb{S}(r) = \emptyset$. On sait aussi qu'un sommet v peut avoir un ancêtre spécial dans au plus $r - 1$ sous-arbres de T . En effet, si r sous-arbres de T contenaient chacun un ancêtre d'un sommet v , alors il existerait r chemins disjoints entre la racine r de T et v passant par chacun des ancêtres dans T . Ce qui constituerait un mineur $K_{2,r}$ dans G ; ce qui est absurde par hypothèse.

Maintenant, nous allons expliquer comment implémenter le champ P . En effet, pour chaque sommet x , le champ P contient les numéros de ports traversés depuis tout sommet spécial y pour lequel $x \in \mathbb{S}(y)$ et x n'appartient pas au $2r$ premiers ports de l'arbre T'_y .

Posons $P = \{p_1, \dots, p_t\}$, avec p_i pour $1 \leq i \leq t$ le i^{me} numéro de port traversé depuis le i^{me} sommet spécial y pour lequel $x \in \mathbb{S}(y)$ et x n'appartient pas au $2r$ premiers ports de l'arbre T'_y . Par conséquent, on a $|P| = \log p_1 + \dots + \log p_t$.

Pour calculer la valeur de $|P|$, nous allons calculer la valeur de chacun des ports p de P . Nous allons aussi montrer que le nombre de ports à stocker dans P ne peut pas dépasser $\log n$, c'est-à-dire $t \leq \log n$.

Le champ P contient l'ensemble des ports à traverser depuis chacun des sommets spéciaux de l'ensemble $Y = \{y_1, \dots, y_t\}$ qui sont tels que $\forall y \in Y, x \in \mathbb{S}(y)$ et que x n'est pas parmi les $2r$ sous-arbres de T'_y les plus lourds. Soit $V = \{y'_1, \dots, y'_t\}$ l'ensemble des voisins respectifs de $y_1, \dots, y_t$ tels que $x \in T'_{y'_i}$ pour $1 \leq i \leq t$.

Soit p_1 le premier port de P . Ce port est donc celui traversé pour arriver en x depuis le premier minimum spécial y_1 pour lequel x n'utilise aucun des $2r$ premiers ports. Le port p_1 est associé à l'arête (y_1, y'_1) .

On sait que le sommet spécial y_1 est le premier sommet pour lequel x doit stocker le port de sortie. Posons C_1 la composante connexe engendrée par $G \setminus \{y_1 \cup V(y_1)\}$ contenant x . Le poids $|T'_{y'_1}|$ du sous-arbre $T'_{y'_1}$ est déterminé par le nombre de sommets spéciaux z tels que $\mathbb{S}(z) \cap T'_{y'_1} \neq \emptyset$. Ces sommets spéciaux pouvant être comptés dans plusieurs autres sous arbres de la même composante connexe C_1 , alors on a : $|T'_{y'_1}| \leq rn/p_1$. En effet, le nombre d'arbres dans une composante connexe est borné par r .

$$\text{Donc } p_1 \leq rn/|T'_{y'_1}|.$$

$$\text{De plus on a, } p \geq 2r. \text{ Donc } |T'_{y'_1}| \leq n/2.$$

Le port p_2 correspond à un sous-arbre en y_2 enraciné en y'_2 . Posons C_2 la composante connexe engendrée par $C_1 \setminus \{y_2 \cup V(y_2)\}$ contenant x . Le sous-arbre $T'_{y'_2}$ contient des sommets du sous-arbre $T'_{y'_1}$ présenté plus haut. Les sommets spéciaux couverts par $T'_{y'_1}$ pouvant être comptabilisés dans d'autres sous-arbres de la même

composante connexe C_2 , alors on a : $|T'_{y'_2}| \leq r|T'_{y'_1}|/p_2$.

Donc $p_2 \leq r|T'_{y'_1}|/|T'_{y'_2}|$.

Par ailleurs, on a $p_2 \geq 2r$ et $|T'_{y'_1}| \leq n/2$. Donc $|T'_{y'_2}| \leq n/2^2$.

Supposons que pour tout port p_i , pour $3 \leq i \leq t-1$, on a $|T'_{y'_{i-1}}| \leq r|T'_{y'_{i-2}}|/p_i$. Montrons que c'est aussi vrai pour le port p_t . En effet, on sait que le sommet spécial y_t est le t^{eme} sommet pour lequel x doit stocker le port de sortie. Le poids $|T'_{y'_t}|$ de $T'_{y'_t}$ est donné par le nombre de sommets spéciaux z tel que $\mathbb{S}(z) \cap T'_{y'_t} \neq \emptyset$. Ces sommets spéciaux sont évidemment dans $T'_{y'_{t-1}}$. Puisque ils peuvent aussi être dans d'autres sous-arbres de la même composante connexe que $T'_{y'_t}$ et que parmi tous les autres sous-arbres, $T'_{y'_t}$ est le p_t^{eme} sous-arbre de poids fort, alors on a $|T'_{y'_t}| \leq r|T'_{y'_{t-1}}|/p_t$. La propriété est vérifiée pour le port p_t .

Donc $p_t \leq r|T'_{y'_{t-1}}|/|T'_{y'_t}|$.

De plus on sait que $p_t \geq 2r$, alors on a $|T'_{y'_t}| \leq n/2^t$.

Or le sous-arbre $T'_{y'_t}$ ne contient que le sommet y_t comme sommet spécial, alors on a $|T'_{y'_t}| = 1$. Ce qui implique, $1 \leq n/2^t$. Par conséquent, on a $t \leq \log n$.

En outre, vu que $p_1 \leq rn/|T'_{y'_1}|$, $p_2 \leq |T'_{y'_1}|/|T'_{y'_2}|$ et que pour tout entier $3 \leq i \leq t$ on a $p_i \leq |T'_{y'_{i-1}}|/|T'_{y'_i}|$, alors on a $\prod_{i=1}^t p_i \leq r^t n$.

Ce qui équivaut à, $\log(\prod_{i=1}^t p_i) = \log(p_1) + \log(p_2) + \dots + \log(p_t) \leq \log(r^t n) = t \log r + \log n$.

Par ailleurs, on sait que $t \leq \log n$. Par conséquent, on a $\log(\prod_{i=1}^t p_i) \leq (1 + \log r)(\log n)$.

Nous avons donc montré que pour tout sommet x , $|P|$ est en $O(\log r \log n)$ bits. Par conséquent, $ADRESSE(x)$ est de $O(\log r \log n)$ bits.

4.3.2 Routage compact de plus courts chemins dans les graphes t -feuilletés

Dans cette partie, nous nous intéressons aux graphes t -feuilletés. Un graphe t -feuilleté est l'union de graphes planaire-extérieurs connexes tel que t est le nombre de sommets avec leurs multiplicités appartenant à la fois à plusieurs sous-graphes.

Nous avons en fait mis en place un schéma de routage de plus courts chemins avec des tables de $O(t \log n)$, bits par sommets, des adresses et des en-têtes de $O(\log n)$ bits. Le temps de décision est de $O(\log t)$ par sommet.

Il est important de noter que le schéma que nous présentons est *universel*, il s'applique à tout graphe en choisissant une valeur adéquate pour t . Pour le cas

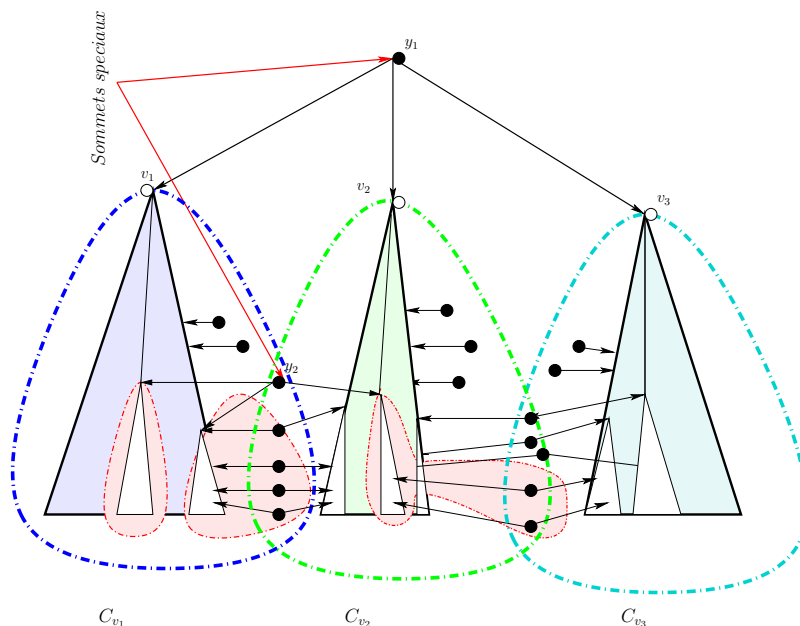


FIG. 4.2: Construction des tables de routage et adresse.

planaire (et de genre borné), il existe un algorithme quadratique pour calculer la décomposition en cellules [Fre91]. D'une certaine façon ce schéma universel exploite la nature géométrique de la topologie sous-jacente.

4.3.2.1 Préliminaires

Une *cellule* d'un graphe G est le plongement d'un sous-graphe connexe de G sur le plan sans croisement d'arête où tous les sommets bordent la face extérieure (cf. figure 4.3(a)). Donc chaque cellule d'un graphe est un graphe planaire extérieur connexe.

Une *couverture* d'un graphe G est un ensemble de sous-graphes de G tel que chaque arête de G soit contenue dans au moins un sous-graphe. On appelle *attachement* tout sommet appartenant à plusieurs cellules. Le *nombre d'attachements* d'une couverture est la somme, sur toutes les cellules, du nombre d'attachements des cellules. Finalement, un graphe *t-feuillet* est un graphe connexe possédant une couverture en cellules de nombre d'attachements au plus t (cf. figure 4.3 (b)).

Les graphes planaires extérieurs connexes sont donc 0-feuillets, et réciproquement. Tout graphe connexe à m arêtes est $2m$ -feuillet, chaque cellule correspondant à une arête et le nombre total d'attachements à la somme des degrés.

Une *partition t-feuillet* d'un graphe G est une partition des arêtes de G en cellules de nombre d'attachements au plus t . Il est facile de voir que tout graphe t -feuillet possède une partition t -feuillet, car supprimer une arête partagée par plusieurs cellules d'une couverture reste une couverture cellulaire dont le nombre d'attachements n'est pas plus grand.

Un *segment* est un ensemble de sommets consécutifs d'une cellule obtenu en parcourant les sommets le long de la face extérieure.

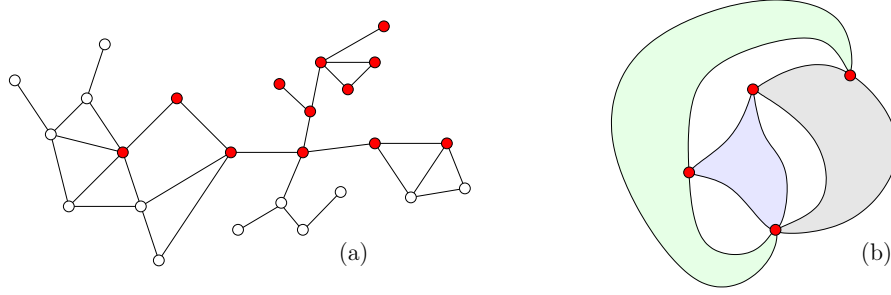


FIG. 4.3: (a) Une cellule où les sommets colorés forment un segment. (b) Trois cellules formant un graphe 9-feuillet.

Pour un arbre T enraciné en x , on note T_u le sous-arbre de T enraciné en u induit par les descendants de u dans T . Un résultat dû à Frederickson et Janardan [FJ88] est que si T enraciné en x couvre un graphe planaire extérieur (donc une cellule), alors pour tout voisin u de x , T_u est un segment.

4.3.2.2 L'algorithme de routage

Considérons une partition t -feuillet d'un graphe G connexe à n sommets.

Pour tout sommet x de G , soit $\text{ATTACH}(x)$ une table de routage construite à partir d'un arbre de plus courts chemins couvrant G enraciné en x dans laquelle seuls les sommets descendants d'au moins un attachement sont préservés. Donc, pour toute destination y , $\text{ATTACH}(x)[y]$ donne le numéro de port de l'arête menant vers y si dans l'arbre choisi par x le chemin utilise au moins un attachement.

Pour toute cellule C de la partition, on choisit un arbre enraciné R^C couvrant² C , de plus courts chemins dans C . Notons que R^C n'est pas forcément un arbre de plus courts chemins de G . Pour tout sommet x de C , on note $\text{TABLECELL}(x, C)$ une table de routage construite à partir d'un arbre enraciné en x couvrant C et de plus courts chemins dans C dans lequel on supprime les sous-arbres issus des voisins de x ne contenant que des descendants de x dans R^C . Donc, pour toute destination $y \notin R_x^C$, $\text{TABLECELL}(x, C)[y]$ donne le numéro de port de l'arête menant vers y .

L'algorithme de routage présenté ci-dessous est élémentaire, l'originalité venant de son implémentation.

Routage de $x \in G$ vers $y \in C$:

1. si $y \in \text{ATTACH}(x)$, retourner $\text{ATTACH}(x)[y]$, sinon
2. si $y \in R_x^C$, router dans l'arbre R^C , sinon
3. retourner $\text{TABLECELL}(x, C)[y]$.

Montrons que le routage est de plus courts chemins. Notons qu'il suffit de montrer que la première étape du routage en x retourne une arête sur un plus court chemin de x à y . C'est évidemment vrai dans le cas 1, si $y \in \text{ATTACH}(x)$. Si $y \notin \text{ATTACH}(x)$

² R^C existe car C est connexe.

cela signifie qu'il existe un plus court chemin entre x et y complètement contenu dans la même cellule, disons C , tout chemin entre deux cellules devant sinon intersecter un attachement. Il suit qu'un plus court chemin entre x et y dans C l'est aussi dans G . Donc par construction de l'arbre R^C , le routage dans le cas 2 est selon un plus court chemin. Finalement, dans le cas 3 le routage est aussi selon un plus court chemin dans C , donc dans G .

4.3.2.3 Implémentation

Il a été démontré dans [FG01b, TZ01] que tout arbre à n sommets possède un schéma de routage utilisant des tables, des en-têtes et des adresses de $O(\log n)$ bits. Ce schéma permet également de déterminer si la destination est un descendant du sommet courant, c'est-à-dire si la route ne passe pas par le père. Dans ce chapitre, on suppose que les numéros de ports de x peuvent être permutés dans d'intervalle $[1, \deg(x)]$ afin d'optimiser la table de routage. Dans le cas où les numéros de ports ne peuvent être permutés, alors la taille des étiquettes $\lambda(x, R)$ augmente jusqu'à $O(\log^2 n / \log \log n)$.

Pour tout arbre R , on notera $\lambda(x, R)$ la concaténation de l'adresse et de la table de x de ce schéma de routage appliqué à R si bien que le routage de x à y peut être déterminé par la seule donnée du couple $(\lambda(x, R), \lambda(y, R))$.

Tout sommet x de G possède un indice $\in \{1, \dots, n\}$ obtenu comme suit : les sommets d'attachements sont numérotés en premier dans un ordre quelconque, les autres sommets étant numérotés consécutivement selon un parcours bordant la face extérieure de chaque cellule.

L'adresse du sommet x est composée du triplé $(i, C, \lambda(x, R^C))$, où i est l'indice de x et C une cellule arbitraire contenant x . Les en-têtes sont simplement composées des adresses, qui sont donc sur $O(\log n)$ bits.

La taille des tables de routage du sommet x est donnée par les tables $\text{ATTACH}(x)$ et $\text{TABLECELL}(x)$, ainsi que $\lambda(x, R^C)$ pour toutes les cellules C contenant x . Il y en a plusieurs si x est un attachement. Il faut aussi stocker une table indiquant la plage des numéros de ports utilisés pour chaque cellule C contenant x . Notons que pour $t \neq 0$ il ne peut avoir plus de cellules que le nombre d'attachements. Autrement dit, x appartient à au plus t cellules. Donc les plages de degrés ainsi que les données $\lambda(x, R^C)$ ne coûtent que $O(t \log n)$ pour la table de x . Reste à évaluer les tailles des tables $\text{ATTACH}(x)$ et $\text{TABLECELL}(x)$.

Les deux tables sont implémentées de manière similaire en compactant toutes les destinations utilisant la même arête (c'est-à-dire le même numéro de port) sous forme d'intervalles d'entiers consécutifs. Pour cela on utilise l'indice des sommets. L'appartenance de y à un segment X donné se fait simplement en testant si l'indice de y est contenu dans un intervalle d'entier (modulo $|C|$) qui code X .

À tout sous-ensemble X de sommets d'une cellule, on associe $\text{seg}(X)$ le plus petit nombre de segments dont l'union est X . Nous avons le résultat clé suivant :

Lemme 4.15 *Soit C une cellule de G , x un sommet de G , et T un arbre de G enraciné en x et couvrant C . Soit A l'ensemble des attachements de C distincts de x*

et soit U l'ensemble des voisins u de x tel que $T_u \cap A \neq \emptyset$. Alors $\sum_{u \in U} \text{seg}(T_u \cap C) \leq 2|A|$.

Autrement dit, la partie de $\text{ATTACH}(x)$ concernant la cellule C peut se compacter en $O(|A|)$ intervalles d'indices (le nombre n'étant pas exactement $2|A|$ car les attachements sont numérotés avant tous les autres sommets et peuvent ainsi ajouter des intervalles). Au total, $\text{ATTACH}(x)$ s'implémente avec $O(t)$ intervalles d'indices, la somme des attachements de G étant de t par définition.

La recherche de $y \in \text{ATTACH}(x)$ peut être effectuée par une recherche dichotomique en temps $O(\log |A|) = O(\log t)$ en rangeant les intervalles par cellules, la cellule C de y étant codée dans l'adresse de y .

Notons que si un graphe G est t -feuillet, alors il existe une partition $O(t)$ -feuillet avec au plus quatre attachements par cellules. Ce qui peut améliorer la latence du routage de $O(\log t)$ à $O(1)$.

Démonstration du lemme 4.15.

Soit une représentation des segments par des sommets ordonnés sur un cycle. On ajoute une arête entre deux segments si dans l'arbre il existe une corde reliant ces deux segments. On supprime les arêtes doubles. Soit $\mathcal{C}$ le graphe ainsi obtenu. Un exemple d'illustration est donné sur la figure 4.4. Vu que les segments sont couverts par l'arbre T , alors toute composante connexe de $\mathcal{C}$ représente un ensemble de segments appartenant à un même sous arbre de T . En effet, les arêtes de $\mathcal{C}$ sont des arêtes de T . L'existence d'un contre exemple conduirait à l'existence d'un cycle dans T ; ce qui est absurde. Ainsi chaque composante connexe induite par un ensemble de segments reliés par des cordes produit des régions du cycle. Sur l'exemple de la figure 4.4 chaque composante connexe est représentée par une couleur. Les régions $\mathbb{R}_1, \mathbb{R}_2, \mathbb{R}_3, \mathbb{R}_4$ sont induites par les cordes.

Soit C l'ensemble des composantes connexes de $\mathcal{C}$ et $R_1, R_2, \dots, R_s$ les régions de $\mathcal{C}$ produites par les composantes connexes de C .

Pour chaque région R , soit $\text{Card}(\text{seg}_{\text{int}}(R))$ le nombre de segments internes de R . Les segments internes de R sont couverts uniquement par R . Ainsi, deux régions R_1, R_2 sont telles que $\text{seg}_{\text{int}}(R_1) \cap \text{seg}_{\text{int}}(R_2) = \emptyset$. De plus pour chaque région R , le nombre de segments internes de R est non nul. En effet, si une région R avait un nombre de segment interne nul, alors les extrémités de la chaîne qui délimite R est constitué de deux segments consécutifs et formeraient donc un seul segment; ce qui contredit le fait que les segments sont compactés au maximum. Donc toute région contient au moins un segment interne à cette région.

On a : $\text{Card}(\text{seg}(G)) = |V(C)| + \sum_{i=1}^s \text{Card}(\text{seg}_{\text{int}}(R_i))$.

Dans un cycle avec m arêtes il y a exactement $m + 1$ régions; car chaque arête reliant uniquement deux sommets du cycle crée par la même occasion une région. Donc, on a $s = |V(\mathcal{C})|$.

En outre, toutes les régions sont ouvertes. En effet une région fermée induirait un cycle avec les arête de l'arbre et donc conduirait à l'existence d'un cycle dans l'arbre; ce qui est absurde.

Si une région ouverte R contient une ou plusieurs composantes connexes, on dit

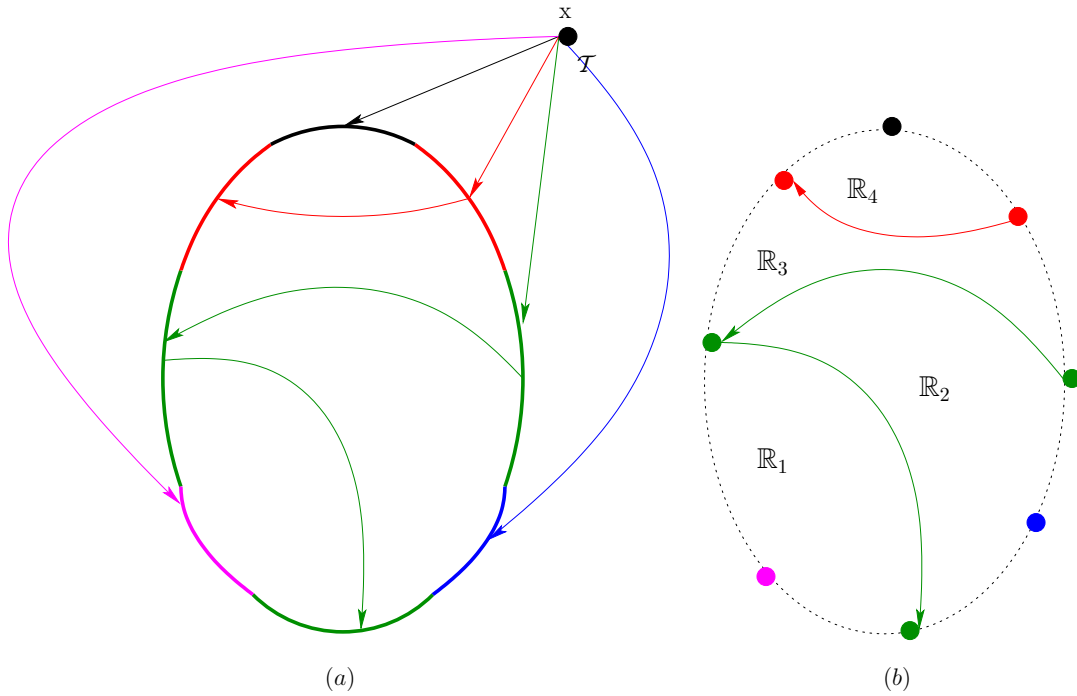


FIG. 4.4: (a) Une cellule C et un arbre $\mathcal{T}$ couvrant C enraciné en x où chaque couleur représente un sous arbre de $\mathcal{T}$ et tout ensemble de sommets consécutifs sur la face extérieure de C est représenté par un segment de même couleur que le sous arbre qui le contient. (b) La représentation des segments de C par des sommets ordonnés sur un cercle et le région engendrées par les cordes de $C \cap \mathcal{T}$.

que R est décomposable. En effet, vu que chaque composante connexe produit au moins une région, alors si une région R contient une composante connexe C alors C décompose la région R en sous régions. Une région non décomposable est une région qui ne contient aucun composante connexe.

Les segments strictement internes à une régions sont connecté dans l'arbre $\mathcal{T}$ par une arête qui n'est pas une corde. En effet toutes les cordes sont dans $\mathcal{C}$. Donc cette arête induit au moins un sommet d'attachement dans ce segment.

Donc on a $\sum_{i=1}^s \text{Card}(\text{seg}_{\text{int}}(R_i)) \leq a$ avec a les nombre d'attachements de la cellule.

Par ailleurs, on a $s = |V(C)|$ et chaque région contient au moins un attachement, alors on a le nombre de régions $S \leq a$.

$$\text{D'où } \text{Card}(\text{seg}(G)) = |V(C)| + \sum_{i=1}^S \text{Card}(\text{seg}_{\text{int}}(R_i)) \leq s + a \leq 2a. \quad \square$$

Le théorème 4.3 nous indique qu'en fait $\text{TABLECELL}(x)$ peut être implémentée en $O(\log n)$ bits.

L'adresse de y contient C et $\lambda(y, R^C)$. Donc en temps constant le sommet x peut décider si $y \in R_x^C$ et le cas échéant calculer la route à l'aide de $\lambda(x, R^C)$. Nous avons

vu que la recherche de y dans $\text{ATTACH}(x)$, étant donné C , en temps $O(\log k)$, tous les autres tests étant constants. Nous avons donc montré le théorème 4.4.

4.4 Conclusion

Nous avons mis en place un schéma de routage de plus courts chemins pour les graphes non valués planaire-extérieurs. Ce schéma de routage utilise des tables de routage, des adresses et des en-têtes de $O(\log n)$ bits.

Nous avons donné deux généralisations de ce résultat, l'une aux graphes valués (k, r) -cellulaires et l'autre aux graphes valués t -feuilletés.

Nous avons montré que les graphes (k, r) -cellulaires valués admettent des propriétés similaires à celles des graphes planaire-extérieurs. Grâce à ces propriétés, nous avons montré que les graphes (k, r) -cellulaires admettent de la même manière que pour les graphes planaire-extérieur un schéma de routage de plus courts chemins avec des tables de routage de $O(kr \log n)$ bits, des adresses et des en-têtes de taille $O(\log r \log n)$. La combinaison de ce résultat avec le fait que les graphes sans mineurs $K_{2,4}$ sont $(2, 4)$ -cellulaires nous permet de donner un schéma de routage de plus courts chemins dans les graphes valués sans mineur $K_{2,4}$. Ce schéma de routage utilise des tables de routage, des adresses et des en-têtes de $O(\log n)$ bits. Les seuls graphes précédemment connus admettant un tel schéma de routage sont les arbres.

Nous pouvons remarquer que le fait que ces graphes excluent $K_{2,r}$ comme mineur se révèle fondamental dans notre schéma. La question qu'on pourrait se poser est de savoir si un schéma de routage similaire pouvait être mis en place pour toute famille de graphes excluant un mineur H planaire ayant k sommets.

Par ailleurs, nous avons montré que les graphes t -feuilletés admettent un schéma de routage de plus courts chemins avec des adresses et en-têtes de $O(\log n)$ bits et des tables de $O(k \log n)$ bits. De plus, le temps de décision est de $O(\log k)$.

Notons que c'est une propriété combinatoire qui est au cœur de ce schéma, en particulier la décomposition cellulaire n'impose aucune restriction sur la métrique des cellules : la distance entre deux sommets d'une cellule est sans relation avec celle dans le graphe en entier. Il serait intéressant d'étendre le schéma à d'autres graphes, et particulier aux graphes de largeur arborescente bornée, mais la taille des séparateurs ne semble pas suffisante pour être exploité dans le cadre du routage.

Conclusion et perspectives

Dans cette partie, nous allons succinctement faire un rappel de l'ensemble des résultats obtenus, et ensuite présenter les perspectives de recherche ouvertes dans cette thèse.

Conclusion

La première partie de cette thèse se subdivise en trois chapitres. Le premier chapitre est constitué des généralités sur les graphes. Dans le deuxième chapitre, nous avons présenté un algorithme polynomial qui, pour tout graphe planaire de longueur arborescente ℓ , donne une décomposition arborescente $\mathcal{T}$ dans laquelle chaque sac a au plus 12 plus courts chemins de longueur ℓ . En effet, chaque sac de $\mathcal{T}$ possède au plus 6 chemins de longueur $\leq 3\ell/2$, composés chacun d'au plus deux plus courts chemins. Ce résultat met donc en évidence une relation entre la largeur arborescente et la longueur arborescente de tout graphe planaire généralisant la relation entre la largeur arborescente et le diamètre des graphes planaires.

Dans le troisième chapitre, nous avons présenté un algorithme de complexité linéaire qui pour tout graphe, 2-connexe soit exhibe un mineur $K_{2,4}$ soit trouve deux sommets u, v dont leur suppression engendre un graphe planaire-extérieur. En d'autres termes, nous avons donc montré que pour tout graphe sans mineur $K_{2,4}$, nous pouvons trouver en temps linéaire deux sommets u, v dont leur suppression engendre un graphe planaire-extérieur. Grâce à ce résultat, nous montrons que les graphes sans mineur $K_{2,4}$ sont de largeur arborescente au plus 4. Cette borne, qui est optimale, améliore ainsi le résultat de Bodlaender *et al.* qui était 6. Nous montrons également que ce résultat ne peut pas être étendu trivialement à tout graphe planaire sans mineur $K_{2,5}$. En effet, nous avons présenté un algorithme qui pour tout entier k , construit un graphe planaire 2-connexe sans mineur $K_{2,5}$ dans lequel il faut enlever au moins k sommets pour le rendre planaire-extérieur.

Par ailleurs, nous avons montré dans ce chapitre que si H est un graphe dessi-nable sur une grille $p \times q$, alors tout graphe G sans mineur H , est de largeur arbores-cente $O(p^{3/2}\sqrt{q})$. En conséquence de ce résultat, nous montrons que les graphes sans mineur $K_{2,r}$ sont de largeur arborescente $O(\sqrt{r})$. Ce qui améliore la borne actuelle due à Thilikos, qui est $r + 2$.

La deuxième partie de cette thèse est constitué par un seul chapitre et elle est dédié au routage compact de plus courts chemins. Nous commençons par donner un schéma de routage de pour les graphes non valués planaire-extérieurs. Ce schéma de routage utilise des tables de routage, des adresses et des en-têtes de $O(\log n)$ bits.

C'est un résultat qui est assez simple mais son importance est de donner une idée plus claire sur la démarche qui est utilisée car c'est cette même démarche qui sera utilisée dans la construction des autres schémas de routage. En effet, nous avons donné deux généralisations de ce schéma : l'une est une généralisation aux graphes valués (k, r) -cellulaires et l'autre aux graphes non valués k -cellulaires.

Nous avons montré que les graphes (k, r) -cellulaires valués admettent des propriétés similaires à celles des graphes planaire-extérieurs. Grâce à ces propriétés, nous avons montré que les graphes (k, r) -cellulaires admettent de la même manière que pour les graphes planaire-extérieurs un schéma de routage de plus courts chemins avec des tables de routage de $O(kr \log n)$ bits, des adresses et des en-têtes de taille $O(\log r \log n)$. La combinaison de ce résultat avec le fait que les graphes sans mineurs $K_{2,4}$ sont $(2, 4)$ -cellulaires nous permet de donner un schéma de routage de plus courts chemins dans les graphes valués sans mineur $K_{2,4}$. Ce schéma de routage utilise des tables de routage, des adresses et des entêtes de $O(\log n)$ bits. Les seuls graphes connus admettant un tel schéma de routage sont les arbres.

Nous pouvons remarquer que le fait que ces graphes excluent $K_{2,r}$ comme mineur se révèle fondamental dans notre schéma. La question qu'on pourrait se poser est de savoir si un schéma de routage similaire pourrait être mis en place pour toute famille de graphes excluant un mineur H planaire ayant k sommets.

Par ailleurs, nous avons montrés que les graphes k -cellulaires admettent un schéma de routage de plus courts chemins avec des adresses et en-têtes de $O(\log n)$ bits et des tables de $O(k \log n)$ bits. De plus, le temps de décision est de $O(\log k)$.

Rappelons que c'est une propriété combinatoire qui est au cœur de ce schéma, en particulier la décomposition cellulaire n'impose aucune restriction sur la métrique des cellules : la distance entre deux sommets d'une cellule est sans relation avec celle dans le graphe en entier.

Perspectives

Dans cette partie, nous allons poser sous forme de questions l'ensemble des points ouverts dans cette thèse, et qui seront l'objet de réflexions dans un futur proche.

Question 1 : La question qu'on pourrait se poser en premier est de savoir si tout graphe de genre bornée et de longueur arborescente ℓ admet une décomposition arborescente dans laquelle chaque sac contient un nombre borné de plus courts chemins de longueur au plus $O(\ell)$.

Question 2 : Afin de se rapprocher de la structure des graphes sans mineur K_6 qui est toujours ouvert, on pourrait se poser la question de savoir ce qu'on peut dire sur la structure des graphes excluant $H = K_{2,4} \cup \{e\}$ comme mineur ? La preuve du chapitre 3 peut être adaptée pour répondre à cette question.

Question 3 : On pourrait aussi se poser la question de savoir quelle est la structure des graphes sans mineur $K_{2,r}$ avec $r > 4$?

Question 4 : On pourrait aussi s'intéresser à la famille $\mathcal{F}$ de graphes la plus large possible, telle que pour tout graphe $H \in \mathcal{F}$ et pour tout graphe G sans mineur H on a la largeur arborescente de G qui est au plus $\sqrt{|V(H)|}$.

Question 5 : Pouvons nous router par de plus courts chemins avec des adresses, des en-têtes et des tables de $O(\log n)$ bits dans les graphes de largeur arborescente 2? Quelle est in finie la propriété structurelle des graphes ayant n sommets et permettant un routage de plus courts chemins avec des tables de routage et des adresses de $O(\log n)$ bits? Pouvons nous étendre ce résultat aux graphes planaire-extérieurs de genre g ?

Index

Symbols	
• $\{1, \dots, n\}$	110
• $(h(u), h(v))$	13
• (x, y)	14
• C	53
• $C^+[u, v]$	52
• G	53
• H_x	85
• $Id_1(x)$	100
• $Id_2(x)$	100
• K_4	16
• K_5	12
• K_6	50
• K_7	50
• K_n	12
• $K_{2,3}$	16
• $K_{2,4}$	15, 84
• $K_{3,3}$	12
• $L \cup R$	53
• L, M, R	53
• M	53
• $M[u, v], M]u, v[$	52
• $N(x)$	86, 87
• $O(n)$	52
• P_y	102
• P_{10}	15
• R	53
• S_0	17
• S_1	17
• $V(G)$	12
• $V(H)$	12
• V_8	49
• V_8	50
• X	91
• $\text{IN}(C)$	53
• $\text{IN}(L \cup M)$	53
• $\text{IN}(M \cup R)$	53
• $\text{IN}(M \cup R)$	53
• A	53
• B	53
• $\Omega(x)$	87
• $\Omega_1(x)$	87
• $\Omega_2(x)$	87
• BFS	14
• DFS	14
• $\diamond(X, Y)$	32
• ℓ	20
• $\in$	110
• $\lambda(x, R)$	110
• $\mathbb{R}(x)$	97
• $\mathbb{R}^2$	17
• $\mathbb{R}_1, \mathbb{R}_2, \mathbb{R}_3, \mathbb{R}_4$	111
• $\mathbb{S}(x)$	97
• $\mathbb{S}_g$	17
• $\mathbb{U}(x)$	97
• $\mathbb{R}^2$	18
• $\text{nca}_T(u, v)$	14
• ϕ	57
• $\phi(L, M, R)$	57
• $\mathcal{F}$	52
• $\mathcal{T}$	21, 49
• $\mathcal{T}_x$	86
• $\sigma :$	18
• $\text{ATTACH}(x)$	109
• $\text{ATTACH}(x)[y]$	109
• $\text{TABLECELL}(x, C)$	109
• $\text{TABLECELL}(x, C)[y]$	109
• $\tilde{G}$	85
• $\tilde{g}$	17
• $\text{tw}(G)$	52
• $ab\text{-séparateur}$	11
• $f(r)$	52

- g 17
 - $i, C, \lambda(x, R^C)$ 110
 - k_G 11
 - k_H 50
 - r 50, 51
 - $p \times q$ 52
 - $ADRESSE-(x)$ 100
 - $ROUTE$ 86
 - $TABLE-H_x$ 86
 - $TABLE-\mathbb{R}(x)$ 99
 - $TABLE-\mathbb{S}(x)$ 99
 - $TABLE-\mathbb{U}(x)$ 100
 - $TABLE-\mathcal{T}_x$ 86
- A**
- adjacents 10
 - adresse 83
 - anses 17
 - anses tordues 17
 - apex|hyperpage 49
 - arête 9
 - arborescence 13
 - arbre 13
 - arbre couvrant 14
 - arbre enraciné 13
 - attachement 108
- B**
- biparti complet 12
- C**
- cellule d'un graphe 108
 - chemin 10
 - conjecture d'Hadwiger 50
 - conjecture de Jorgensen 50
 - couverture d'un graphe 108
 - cycle 10
- D**
- Décomposition arborée 26
 - décomposition arborescente 21
 - décomposition arborescente connexe
22
 - demi-diamant 32
 - descendant 14
 - diamètre 10
 - diamant 32
- distance 10
- E**
- en-tête de message 83
 - espace topologique 17
 - extrémités 9
- F**
- f-propriété d'intervalle 91
 - feuille 14
 - fonction de numérotation 91
 - fonction de routage 83
 - forêt 14
 - frères 14
 - frontière 17
- G**
- graphe k -connexe 11
 - graphe t -feuillet 85
 - graphe bi-connexe 11
 - graphe planaire-extérieur 16
 - graphe (k, r) -cellulaire 84, 90
 - graphe 2-connexe 11
 - graphe h -almost embeddable 49
 - graphe t -feuillet 107
 - graphe (k, r) -cellulaire 91
 - graphe biparti 12
 - graphe complet 12
 - graphe connexe 11
 - graphe de Wagner 49
 - graphe plan 18
 - graphe planaire-extérieur 84
 - graphe sans mineur K_r 50
 - graphes planaires sans mineur $K_{2,r}$
50
 - graphes sans mineur K_5 50
 - graphes sans mineur K_6 50
 - graphes simples 9
 - grille $K_{p \times q}$ 10
- H**
- homomorphisme de graphes 12
- L**
- longueur des routes 83
- M**
- mineur H dessinable sur une grille
 $p \times q$ 71

• mineur de graphe.....	14	• sous-graphe induit.....	13
• Mineur topologique.....	16	• structure des graphes sans mineur H	49
N			
• nœud.....	21	• subdiviser une arête.....	10
• nœud interne.....	14	• super-graphe.....	13
• nombre chromatique.....	50	• surface non-orientable de genre g ..	17
O			
• obstruction.....	16	• surface orientable de genre g	17
P			
• pôles de diamant.....	44	• table de routage.....	83
• père.....	14	• Théorème d'Euler.....	18
• parcours en largeur.....	14	• Théorème de Kuratowski.....	12
• parcours en profondeur.....	14	• Théorème de Wagner.....	15
• parent.....	14	• tree-length.....	21
• partie privée.....	23	• tree-width.....	21
• partition RSU de G	96	V	
• partition t -feuillet.....	108, 109	• voisins.....	10
• Partition de G	96		
• plan Euclidien.....	18		
• plongement de graphe sur une sur-			
face.....	17		
• plongement-LMR d'un graphe...	53		
• plus court chemin.....	10		
• plus courts chemins.....	96		
• problème NP -complet.....	19		
• proche.....	97		
• profondeur.....	14		
• Propriété d'intervalle.....	91		
R			
• racine.....	13		
S			
• séparateur.....	11		
• séparateur minimal.....	11		
• sacs.....	21		
• schéma de routage.....	83		
• schéma de routage universel....	107		
• segment.....	108		
• sillon de diamant.....	32		
• sommet.....	9		
• sommet d'articulation.....	11		
• sous-graphe.....	13		
• sous-graphe couvrant.....	13		

Bibliographie

- [ACP87] Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k -tree. *SIAM Journal on Algebraic and Discrete Methods*, 8 :277–284, 1987.
- [ACPS93] Stefan Arnborg, Bruno Courcelle, Andrzej Proskurowski, and Detlef Seese. An algebraic theory of graph reduction. *J. ACM*, 40(5) :1134–1164, 1993.
- [AGGM06] Ittai Abraham, Cyril Gavoille, Andrew V. Goldberg, and Dahlia Malkhi. Routing in networks with low doubling dimension. In *26th International Conference on Distributed Computing Systems (ICDCS)*. IEEE Computer Society Press, July 2006.
- [AGM06] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. On space-stretch trade-offs : Upper bounds. In *18th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 207–216. ACM Press, July 2006.
- [AGM⁺08] Ittai Abraham, Cyril Gavoille, Dahlia Malkhi, Noam Nisan, and Mikkel Thorup. Compact name-independent routing with minimum stretch. *ACM Transactions on Algorithms*, 3(4) :Article 37, June 2008.
- [AHU72] A. Aho, J. Hopcroft, and J. Ullman. *Design and Analysis of Computer Algorithms*. Addison-Wesley, 1972.
- [AP89] Stefan Arnborg and Andrzej Proskurowski. Linear time algorithms for np-hard problems restricted to partial k -trees. *Discrete Appl. Math.*, 23(1) :11–24, 1989.
- [Arn85] Arnborg. Efficient algorithms for combinatorial problems on graphs with bounded, decomposability-a survey. *BIT Computer Science and Numerical Mathematics*, 25 :2 – 23, 1985.
- [BE97] Hans L. Bodlaender and Joost Engelfriet. Domino treewidth. *J. Algorithms*, 24(1) :94–123, 1997.
- [Ber83] Claude Berge. *Graphs and Hypergraphs*. North-Holland Publishing, 1983.
- [BGKH92] Hans L. Bodlaender, John R. Gilbert, Ton Kloks, and Hjalmtyr Hafsteinsson. Approximating treewidth, pathwidth, and minimum elimination tree height. In *WG '91 : Proceedings of the 17th International Workshop*, pages 1–12, London, UK, 1992. Springer-Verlag.

- [BH98] Hans L. Bodlaender and Torben Hagerup. Parallel algorithms with optimal speedup for bounded treewidth. *SIAM J. Comput.*, 27(6) :1725–1746, 1998.
- [BKK95a] Hans L. Bodlaender, Ton Kloks, and Dieter Kratsch. Treewidth and pathwidth of permutation graphs. *SIAM J. Discret. Math.*, 8(4) :606–616, 1995.
- [BKK95b] Hans L. Bodlaender, Ton Kloks, and Dieter Kratsch. Treewidth and pathwidth of permutation graphs. *SIAM J. Discret. Math.*, 8(4) :606–616, 1995.
- [BLTT97] Hans Leo Bodlaender, Jan van Leeuwen, Richard B. Tan, and Dimitrios M. Thilikos. On interval routing schemes and treewidth. *Information and Computation*, 139 :92–109, November 1997.
- [BM93] Hans L. Bodlaender and Rolf H. Möhring. The pathwidth and treewidth of cographs. *SIAM J. Discret. Math.*, 6(2) :181–188, 1993.
- [BM99] John Boyer and Wendy Myrvold. Stop minding your P’s and Q’s : a simplified $O(n)$ planar embedding algorithm. In *10th Symposium on Discrete Algorithms (SODA)*, pages 140–146. ACM-SIAM, 1999.
- [Bod87] Hans Leo Bodlaender. Dynamic programming on graphs with bounded treewidth. Technical Report RUU-CS-87-22, Department of Information and Computing Sciences, Utrecht University, 1987.
- [Bod88] Hans Leo Bodlaender. Nc-algorithms for graphs of small treewidth. Technical Report RUU-CS-88-04, Department of Information and Computing Sciences, Utrecht University, 1988.
- [Bod90] Hans Leo Bodlaender. Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees. *J. Algorithms*, 11(4) :631–643, 1990.
- [Bod96] Hans Leo Bodlaender. A partial k -arboretum of graphs with bounded treewidth. Technical Report RUU-CS-1996-02, Utrecht University, 1996.
- [Bod98] Hans Leo Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2) :1–45, 1998.
- [Bod05] Hans Leo Bodlaender. Discovering treewidth. In *SOFSEM*, pages 1–16, 2005.
- [Bod06] Hans Leo Bodlaender. Treewidth : Characterizations, applications, and computations. In *WG*, pages 1–14, 2006.
- [CDE⁺08] Victor D. Chepoi, Feodor F. Dragan, Bertrand Estellon, Michel Habib, and Yann Vaxès. Diameters, centers, and approximating trees of delta-hyperbolic geodesic spaces and graphs. In *24st Annual ACM Symposium on Computational Geometry (SoCG)*, page ?, June 2008.

-
- [Chl07] Janka Chlebíková. A characterization of some graph classes using excluded minors. *Periodica Mathematica Hungarica*, 55(1) :1–9, September 2007.
- [CiN95] Marek Chrobak and Shin ichi Nakano. Minimum-width grid drawings of plane graphs. *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes in Computer Science*, 10 :104–110, 1995.
- [CM93] Bruno Courcelle and Mohamed Mosbah. Monadic second-order evaluations on tree-decomposable graphs. *Theor. Comput. Sci.*, 109(1-2) :49–82, 1993.
- [Cou90] Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1) :12–75, 1990.
- [Cou96] Bruno Courcelle. Equivalent definitions of recognizability for sets of graphs of bounded tree-width,. *Structure in Comp. Sci.*, 6 :141–165, 1996.
- [Cow01] Lenore Jennifer Cowen. Compact routing with minimum stretch. *Journal of Algorithms*, 38(1) :170–183, 2001.
- [CRS08] Maria Chudnovsky, Bruce Reed, and Paul D. Seymour. The densest graphs with no $K_{2,t}$ minor. In *Graph Theory Workshop (Carsten Thomassen's 60th birthday)*, August 2008.
- [CV03] Bruno Courcelle and Rémi Vanicat. Query efficient implementation of graphs of bounded clique-width. *Discrete Applied Mathematics*, 131 :129–150, 2003.
- [DF99] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.
- [DG03] Yon Dourisboure and Cyril Gavaille. Tree-decomposition of graphs with small diameter bags. In J. Fila, editor, *2nd European Conference on Combinatorics, Graph Theory and Applications (EUROCOMB)*, pages 100–104, September 2003.
- [DG04a] Yon Dourisboure and Cyril Gavaille. Small diameter bag tree-decompositions. Research Report RR-1326-04, LaBRI, University of Bordeaux 1, 351, cours de la Libération, 33405 Talence Cedex, France, May 2004.
- [DG04b] Yon Dourisboure and Cyril Gavaille. Sparse additive spanners for bounded tree-length graphs. Research Report RR-1316-04, LaBRI, University of Bordeaux 1, 351, cours de la Libération, 33405 Talence Cedex, France, February 2004.
- [DG04c] Yon Dourisboure and Cyril Gavaille. Sparse additive spanners for bounded tree-length graphs. In *11th International Colloquium on Structural Information & Communication Complexity (SIROCCO)*, volume 3104 of *Lecture Notes in Computer Science*, pages 123–137. Springer, June 2004.

- [DG07a] Youssou Dieng and Cyril Gavoille. Routage dans les graphes cellulaires. In *9^{èmes} Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel)*, pages 91–94, May 2007.
- [DG07b] Yon Dourisboure and Cyril Gavoille. Tree-decompositions with bags of small diameter. *Discrete Mathematics*, 307(16) :2008–2029, 2007.
- [DG08] Youssou Dieng and Cyril Gavoille. La structure des graphes sans mineur $K_{2,4}$. In *10^{èmes} Journées Graphes et Algorithmes*, page 9, November 2008.
- [DG09a] Youssou Dieng and Cyril Gavoille. On the tree-width of planar graphs. In Elsevier, editor, *European Conference on Combinatorics, Graph Theory and Applications (EuroComb)*, volume 34C, pages 593–596. Electronic Notes in Discrete Mathematics, September 2009.
- [DG09b] Youssou Dieng and Cyril Gavoille. Routage dans les réseaux (k, r) -cellulaires. In *10^{èmes} Journées Doctorales en Informatique et Réseaux (JDIR)*, pages 7–12, February 2009. Sélectionné pour un numéro spécial de la revue TSI.
- [Dom07] Michael Dom. Compact routing. In *Algorithms for Sensor and Ad Hoc Networks*, volume 4621 of Lecture Notes in Computer Science, pages 187–202. Springer, 2007.
- [DPP90] Hubert DeFraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10 :41–51, 1990.
- [FG01a] Pierre Fraigniaud and Cyril Gavoille. Comment router dans un arbre ? In *3^{èmes} Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel)*, pages 85–88. INRIA, May 2001.
- [FG01b] Pierre Fraigniaud and Cyril Gavoille. Routing in trees. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *28th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2076 of Lecture Notes in Computer Science, pages 757–772. Springer, July 2001.
- [FG02] Pierre Fraigniaud and Cyril Gavoille. A space lower bound for routing in trees. In *19th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2285 of Lecture Notes in Computer Science, pages 65–75. Springer, March 2002.
- [FG06] Jörg Flum and Martin Grohe. *Parametrized Complexity Theory*. Springer, 2006.
- [FHL08] Uriel Feige, MohammadTaghi Hajiaghayi, and James R. Lee. Improved approximation algorithms for minimum-weight vertex separators. *SIAM Journal on Computing*, 38(2) :629–657, May 2008.
- [FJ88] Greg N. Frederickson and Ravi Janardan. Designing networks with compact routing tables. *Algorithmica*, 3(1) :171–190, August 1988.

-
- [Fre91] Greg N. Frederickson. Planar graph decomposition and all pairs shortest paths. *Journal of the ACM*, 38(1) :162–204, 1991.
- [Gav01] Cyril Gavoille. Routing in distributed networks : Overview and open problems. *ACM SIGACT News - Distributed Computing Column*, 32(1) :36–52, March 2001.
- [Had43] H. Hadwiger. Über eine klassifikation der streckenkomplexe, vierteljahrsschr. *Vierteljahrsschr Naturforsch Ges. Zurich*, 88 :133–142., 1943.
- [HM94] Michel Habib and Rolf H. Möhring. Treewidth of cocomparability graphs and a new order-theoretic parameter. *Springer*, 11 :47–60, 1994.
- [Jør94] Leif K. Jørgensen. Contractions to K_8 . *Graph Theory*, 18 :431–448, 1994.
- [Kaw07] Ken-ichi Kawarabayashi. Half integral packing, erdős-posá-property and graph minors. In *18th Symposium on Discrete Algorithms (SODA)*, pages 1187–1196. ACM-SIAM, January 2007.
- [KCFB07] Dmitri Krioukov, KC Claffy, Kevin Fall, and Arthur Brady. On compact routing for the internet. *ACM SIGCOMM Computer Communication Review*, 37(3) :43–52, July 2007.
- [Klo93] Ton Kloks. Treewidth of circle graphs. In *ISAAC '93 : Proceedings of the 4th International Symposium on Algorithms and Computation*, pages 108–117, London, UK, 1993. Springer-Verlag.
- [Kur30] Kazimierz Kuratowski. Sur le problème des courbes gauches en topologie. *Math.*, 15 :271–283, 1930.
- [Lok07] Daniel Lokshtanov. On the complexity of computing treelength. In *32nd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume Lecture Notes in Computer Science. Springer, August 2007.
- [Mah00] John Maharry. A characterization of graphs with no cube minor. *J. Comb. Theory Ser. B*, 80(2) :179–201, 2000.
- [MT92] Jiří Matoušek and Robin Thomas. On the complexity of finding iso- and other morphisms for partial k-trees. *Discrete Math.*, 108(1-3) :343–364, 1992.
- [MT01] Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. The Johns Hopkins university Press, 2001.
- [OOT93] Bogdan Oporowski, James Oxley, and Robin Thomas. Typical subgraphs of 3- and 4-connected graphs. *J. Comb. Theory Ser. B*, 57(2) :239–257, 1993.
- [Pel00] David Peleg. Proximity-preserving labeling schemes. *Journal of Graph Theory*, 33(3) :167–176, 2000.

- [PN06] Fraigniaud Pierre and Nisse Nicolas. Connected treewidth and connected graph searching. In *Proceeding of the 7th Latin American Symposium on Theoretical Informatics (LATIN)*, pages 479–490. Springer LNCS 3887, 2006.
- [PT93] Andrzej Proskurowski and Jan Arne Telle. Practical algorithms on partial k-trees with an application to domination-like problems. *Computer Science*, 709 :610–621, April 1993.
- [PU89] David Peleg and Eli Upfal. A trade-off between space and efficiency for routing tables. *Journal of the ACM*, 36(3) :510–530, July 1989.
- [RS86] Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3) :309–322, 1986.
- [RS95] Neil Robertson and Paul D. Seymour. Graph minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1) :65–110, 1995.
- [RS03] Neil Robertson and Paul D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1) :43–76, 2003.
- [RST93] Neil Robertson, Paul D. Seymour, and Robin Thomas. Hadwiger’s conjecture for K_6 -free graphs. *Combinatorica*, 13(3) :279–361, 1993.
- [RST94] Neil Robertson, Paul D. Seymour, and Robin Thomas. Quickly excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 62(2) :323–348, 1994.
- [Sch90] Walter Schnyder. Embedding planar graphs on the grid. In *1st Symposium on Discrete Algorithms (SODA)*, pages 138–148. ACM-SIAM, January 1990.
- [SSR94] Ravi Sundaram, Karan Sher Singh, and C. Pandu Rangan. Treewidth of circular-arc graphs. *SIAM J. Discret. Math.*, 7(4) :647–655, 1994.
- [ST94] Paul D. Seymour and Robin Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2) :217–241, 1994.
- [Thi99] Dimitrios M. Thilikos. Quickly excluding a $K_{2,r}$ from planar graphs. In Elsevier, editor, *6th Twente Workshop on Graphs and Combinatorial Optimization*, volume 3, pages 189–194. Electronic Notes in Discrete Mathematics, May 1999.
- [Tho04] Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM*, 51(6) :993–1024, November 2004.
- [TZ01] Mikkel Thorup and Uri Zwick. Compact routing schemes. In *13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 1–10. ACM Press, July 2001.

-
- [UY09] Kaori Umezawa and Koichi Yamazaki. Tree-length equals branch-length, 2009.
- [Wag37] K. Wagner. Über eine Eigenschaft der ebenen Komplexe. *Mathematische Annalen*, 114(1) :570–590, December 1937.
- [Wil84] S. G. Williamson. Depth-first search and Kuratowski subgraphs. *Journal of the ACM*, 31(4) :681–693, 1984.

Décomposition arborescente des graphes planaires et routage compact

Résumé : Savoir comment transmettre une information est fondamental dans un réseau. Il est essentiel que chaque entité du réseau soit capable de décider localement, avec sa vue du réseau, du chemin par lequel l'information doit passer. Ainsi, il est souvent utile d'étudier la topologie du réseau, modélisée par un graphe, pour répondre à ces exigences.

Nous nous intéressons dans un premier temps, à la décomposition arborescente des graphes planaires. En effet, comme dans beaucoup de problèmes de graphes, l'étude de la topologie des graphes nous conduit à procéder à une décomposition du graphe afin d'exploiter les propriétés structurelles qui en découlent.

En suite, nous nous sommes aussi intéressés à la structure des graphes qui excluent un mineur H , en particulier le graphe $K_{2,r}$. Ces travaux nous ont permis d'améliorer les bornes actuelles connues sur la largeur arborescente de ces graphes.

Dans la dernière partie, nous abordons le problème du routage compact. Nous nous sommes intéressés aux schémas de routage de plus courts chemins utilisant des adresses, des tables de routage de tailles optimales de $O(\log n)$ bits, où n est le nombre de sommets du graphe. Nous proposons un tel schéma de routage pour une famille de graphes valués contenant les arbres et les graphes planaire-externes.

Mots clefs : Routage compact, graphe planaire, graphe planaire-externe, décomposition arborescente, largeur arborescente, longueur arborescente, graphe sans mineur.

Tree-decomposition of planar graphs and compact routing

Abstract : In a network, it is crucial to know how to construct an efficient routing scheme. It is fundamental for each entity with its local knowledge of the network, to be able to decide on which link to forward messages. Thus, it is important to study the underlying network topology in order to design routing schemes.

In the first part of this thesis, we construct a new tree-decomposition for planar graphs. In fact, as in many graph problems, the study of the graph structure leads to do a tree-decomposition for exploiting structural properties of the graphs.

In the second part, we have studied the structure of H -minor free graphs, in particular when $H = K_{2,r}$. Our results improve upon previous known bounds about the tree-width of $K_{2,r}$ -minor free graphs.

At last, we treat the problem of compact routing scheme. More precisely, we are interested in shortest-path routing schemes that use $O(\log n)$ bits for addresses, headers and routing tables, where n is the number of vertices in the graph. We propose such a routing scheme for a large family of weighted graphs including outerplanar graphs.

Keywords : Compact routing, planar graph, outerplanar-graph, tree-decomposition, tree-width, tree-length, minor free graph.