

Académie de Montpellier
Université Montpellier II
Sciences et Techniques du Languedoc

MÉMOIRE DE STAGE DE MASTER 2

effectué au Laboratoire d'Informatique de Robotique
et de Micro-électronique de Montpellier

Spécialité : **Professionnelle et Recherche unifiée en Informatique**

**Algorithme FPT pour le problème de la
complétion plane à diamètre borné**

par **Clément Requilé**

Date de soutenance : **26 juin 2014**

Sous la direction de **Dimitrios M. Thilikos**

Rapporteurs : **Ignasi Sau** et
Rodolphe Giroudeau

Table des figures

1.1	Un graphe et sa décomposition arborescente.	9
1.2	Une décompositon arborescente optimale de la grille 3×3 , de largeur trois et séparée par l'arête verte ; son ensemble mitoyen est $\{e, f, h\}$	12
1.3	Les mineurs exclus pour les graphes à largeur arborescente au plus trois.	13
1.4	Lasso de taille 4	15
2.1	18
2.2	Le graphe $\Gamma_{7,4}$ dont la base et le toit sont respectivement les cercles en gras de plus petit et de plus grand rayon.	22
2.3	Un sommet d'articulation de connectivité trois remplacé par un cycle.	24
2.4	Carte plane simple sur les cycles d'une face de G	28
2.5	Un graphe planaire extérieur 2-connexe G , son dual faible G_w^* et ses faces simpliciales.	34
2.6	$C + M$ et une 3-partition des arêtes de $C + \Gamma$. Les arêtes rouges sont les arêtes de E^0 , les noires celles de E^s et les vertes celles de E^∞ . Une arête interne de $C + M$ est associée à chemin rouge-vert dans $C + \Gamma$	37
2.7	$C + M$ dont le dual faible est : un chemin (gauche), une étoile (milieu) un arbre ternaire (droite).	37
3.1	$[0, 1]$ -partitions des sommets d'un lasso.	45
3.2	La table stockée dépend de $ X_e $ et de d uniquement.	46
3.3	Nouvelle distance entre x et y induite par la fusion des lassos O_l et O_r	51

Liste des Algorithmes

1	grapheLisse	23
2	sommetLisse	24
3	partitionnement	36
4	jointure	48
5	fusion	48
6	BCP	55

Liste des tableaux

4.1 Résultats	58
-------------------------	----

Table des matières

Introduction	1
1 Préliminaires	2
1.1 Définitions et notations	2
1.1.1 Graphes dans le plan	2
1.1.2 Complexité paramétrée et problèmes de complétion	7
1.1.3 Décompositions de graphes	9
1.2 Largeurs bornées et algorithmes FPT	13
1.2.1 Mineurs	13
1.2.2 Programmation dynamique	14
2 Complétion plane à diamètre borné	17
2.1 Problèmes et résultats	17
2.1.1 Définitions	17
2.1.2 Complexités	17
2.1.3 Variantes	19
2.1.4 Un algorithme FPT paramétré par d et k	20
2.2 Augmentation radiale	21
2.2.1 Grille annulaire	21
2.2.2 Graphe lisse	23
2.2.3 L'augmentation radiale	27
2.3 La (d, k, s, w) -arête colorabilité	32
2.3.1 Définition	32
2.3.2 Équivalence entre BFCP et la (d, k, s, w) -arête colorabilité	33
3 Programmation dynamique sur la (d, k)-arête colorabilité	40
3.1 Préliminaires	41
3.1.1 Décomposition en coupes sphériques	41
3.1.2 (d, k) -configurations	41
3.2 Programmation dynamique	43
3.2.1 Les tables	43
3.2.2 Jointure et fusion	47
3.2.3 Complexité de la programmation dynamique	51
3.3 Un algorithme FPT pour le problème BCP	54
3.3.1 L'algorithme pour BFCP	54
3.3.2 Complexité	55
4 Tableau récapitulatif et pistes de recherche	57
4.1 Améliorations possibles	58
4.1.1 Connecter le graphe	58
4.1.2 La fonction γ	59
4.2 Pistes de recherche	59
4.2.1 Grille annulaire et problèmes de complétion	59
4.2.2 Problèmes ouverts	60
Conclusion	61

Introduction

Modifier un graphe en supprimant ou en ajoutant des arêtes ou des sommets afin qu'il satisfasse certaines propriétés est une opération naturelle en théorie des graphes comme dans de nombreuses applications.

Dans sa thèse, Sharan [Sharan, 2002] étudie un certain nombre d'applications à la génétique de problèmes de modification de graphes, comme l'agencement de séquences génomiques ou la reconstitution de génomes à partir de fragments dispersés.

Dans leur célèbre monographie, Garey et Johnson [Garey and Johnson, 1979] mentionnaient déjà 18 de ces problèmes sur les graphes. Ainsi, beaucoup de problèmes de la théorie des graphes peuvent se voir comme des problèmes de modification de graphes.

Cependant, pour nombre de ces problèmes, aucun algorithme efficace n'est connu pour les résoudre, et certains ont même été prouvés NP-Difficiles.

Parmi tous les problèmes de modification, le premier d'entre eux à avoir été étudié comme tel fut celui de la complétion en un graphe cordal, i.e. dont tous les cycles de longueur au moins quatre admettent une corde. Yannakakis a ainsi montré en 1981 [Yannakakis, 1981] que ce problème était NP-Complet.

Afin de remédier à cette difficulté, plusieurs méthodes ont été développées comme les algorithmes d'approximation, sous-exponentiels ou paramétrés. Ces derniers ont été formalisés pour la première fois par Downey et Fellows dans une série de quatre articles parus entre 1993 et 1995 dont une version étoffée est éditée sous la forme d'un livre [Downey and Fellows, 1999].

Un graphe plan est un plongement sans croisement d'un graphe planaire dans le plan. En 1987, Chung [Chung, 1987] a proposé le problème suivant : étant donné un graphe plan, est-il possible de lui ajouter des arêtes afin de diminuer le diamètre du graphe résultant tout en garantissant que ce dernier reste plan ?

Ce problème est un problème de complétion d'arêtes dans la veine de celui étudié par Yannakakis. Ce problème est en fait NP-COMPLET. Une étude de sa complexité paramétrée s'avère donc nécessaire.

L'objectif de ce mémoire est d'essayer de mener une telle étude. Dans ce but, nous allons tenter de montrer que cette complexité peut dépendre d'autres paramètres que la simple taille de la donnée. Par exemple les deux paramètres dépendant du problème que sont le diamètre final à atteindre d et le nombre k d'arêtes ajoutées lors de la complétion.

Nous donnerons alors pour ce problème un algorithme FPT paramétré par d et k . Cet algorithme repose essentiellement sur deux points. Premièrement, sur le constat que ce problème est équivalent à un autre problème qui lui n'est plus un problème de complétion d'arêtes ni même de modification de graphe en général.

Et enfin, sur une programmation dynamique basé sur une décomposition en coupes sphériques de ce problème équivalent.

1.1 Définitions et notations

1.1.1 Graphes dans le plan

Un peu de théorie des ensembles

Un ensemble est une collection finie d'objets différents deux-à-deux. Un sous-ensemble F d'un ensemble E est une partie de la collection d'objets associée à E . Le cardinal ou taille d'un ensemble E , noté $|E|$, est le nombre d'éléments appartenant à E . L'ensemble des sous-ensembles de taille k d'un ensemble E est noté $\binom{E}{k}$. Ainsi, l'ensemble des parties d'un ensemble E se note 2^E . Pour $k \in \mathbb{N}$, l'ensemble $\{1, \dots, k\}$ sera noté $[k]$. Une partition d'un ensemble X est un ensemble \mathcal{X} de parties de X tel que :

- $X \subseteq \cup_{X' \in \mathcal{X}} X'$ et
- pour chaque $X_1, X_2 \in \mathcal{X}$, $X_1 \cap X_2 = \emptyset$.

Une *relation binaire* sur un ensemble X est un ensemble de couples ordonnés de X . Une relation binaire \leq sur un ensemble X est un *pré-ordre* lorsqu'elle est

- réflexive : $\forall x \in X, x \leq x$ et
- transitive : $x \leq y$ et $y \leq z \implies x \leq z$, pour $x, y, z \in X$.

Un ordre partiel \leq sur un ensemble X est pré-ordre antisymétrique sur X : $x \leq y$ et $y \leq x \implies x = y, \forall x, y \in X$. Un pré-ordre partiel \leq sur un ensemble X est un *bel ordre*, lorsque

- toute suite infinie décroissante d'éléments de X stagne à partir d'un certain rang et
- tout sous-ensemble non vide de X admet un plus petit élément.

Une relation binaire \sim sur un ensemble X est appelée *relation d'équivalence* lorsqu'elle est :

- réflexive,
- symétrique : $x \sim y \implies y \sim x$ et
- transitive.

Cette relation définit une partition de X en classes, que nous appellerons *classes d'équivalences* de X . La *fermeture transitive* d'une relation binaire R est la plus petite relation binaire transitive contenant R , vu en tant qu'ensemble de couples.

Un ordre cyclique sur un ensemble X est une relation ternaire $C \subseteq X^3$, notée $[a, b, c]$ pour $a, b, c \in X$, et satisfaisant les propriétés suivantes :

- cyclicité : si $[a, b, c]$, alors $[b, c, a]$,
- asymétrie : si $[a, b, c]$, alors non $[c, b, a]$,
- transitivité : si $[a, b, c]$ et $[a, c, d]$, alors $[a, b, d]$,
- totalité : si a, b , et c sont distincts, alors soit $[a, b, c]$, soit $[c, b, a]$

Une fonction $f : E \rightarrow F$ de l'ensemble E vers l'ensemble F associe à chaque élément $e \in E$ un élément $f(e) \in F$. C'est la relation binaire $\{(e, f(e)) \mid e \in E\}$. Une fonction $f : E \rightarrow F$ est dite injective lorsque pour tout $e, e' \in E$, $f(e) = f(e') \implies e = e'$. Elle est surjective lorsque $F \subseteq f(E)$ et bijective quand elle est à la fois injective et surjective.

Pour finir, voici une définition qui nous sera particulièrement utile par la suite.

Définition 1 (*Ordres des sous-ensembles avec parité*)

Soient X un ensemble et $q \in \{0, 1\}$, alors nous posons

$$\text{ord}^q(X) = \{\pi \text{ ordre sur } X' \mid \exists X' \subseteq X : X' \neq \emptyset \wedge |X'| \bmod 2 = q\}.$$

Généralités sur les graphes

Un *graphe simple*, non orienté et sans boucle G est la donnée d'un couple $(V(G), E(G))$, où $V(G)$ est un ensemble fini, appelé l'ensemble des *sommets* du graphe, et $E(G) \subseteq \binom{V(G)}{2}$ est un ensemble de couples de $V(G)$, appelé l'ensemble des *arêtes* du graphe. Lorsque $E(G)$ est un ensemble de couples ordonnés, i.e. $E(G) \subseteq V \times V$, le graphe est dit *orienté*. Une *boucle* d'un graphe simple est un couple composé de deux fois le même sommet ; intuitivement, c'est une arête ou un arc partant d'un sommet et revenant à ce même sommet. Dans la suite, G sera un graphe simple, non orienté et sans boucle tel que $|V(G)| = n \in \mathbb{N}$.

Un *sous-graphe* G' de G est un graphe tel que $V(G') \subseteq V$ et $E(G') \subseteq E$. Le sous-graphe de G induit par un ensemble de sommets $V' \subseteq V(G)$, noté $G[V']$, est la donnée du couple (V', E') , où $E' \subseteq E(G)$ et pour tout $a, b \in V'$, $\{a, b\} \in E(G) \implies \{a, b\} \in E'$. De même, le sous-graphe de G induit par un ensemble d'arêtes $E' \subseteq E(G)$, noté $G[E']$, est la donnée du couple (V', E') , où $V' = \{V(G) \cap e \mid e \in E'\}$. Le fait pour un graphe H d'être un sous-graphe de G sera noté $H \leq G$. Et, lorsqu'il n'y a pas d'ambiguïté, le sous-graphe de G dont on a supprimé un sommet $v \in V(G)$ (resp. une arête $e \in E(G)$) sera noté $G - v$ (resp. $G - e$).

Dans un graphe non-orienté G , une arête $e \in E(G)$ est dite *adjacente* ou *incidente* à un sommet $v \in V(G)$ lorsque $v \in e$. De même, deux arêtes sont adjacentes (resp. deux sommets) si elles sont adjacentes à un même sommet (resp. à une même arête). Le *degré d'un sommet* $v \in V(G)$, noté $\delta(v)$, représente le nombre d'arêtes adjacentes à ce sommet, i.e. $\delta(v) = |\{e \in E(G) \mid v \in e\}|$. Le graphe *complet* à n sommets, noté K_n , est le graphe dont chaque sommet est adjacent à tous les autres.

Un *chemin* de G est une suite alternées de sommets et d'arêtes (commençant et terminant par un sommet) du graphe telle que deux éléments consécutifs de la suite sont adjacents. La *longueur* ou *taille* d'un chemin est le nombre d'arêtes de la suite qui lui est associée. La *distance* entre deux sommets de G est la longueur minimale entre tous les chemins du graphe reliant ces deux sommets. Le *diamètre* de G est alors le maximum parmi toutes les distances entre deux sommets du graphe. Un *r -voisinage* d'un sommet $v \in V(G)$, pour $r \in \mathbb{N}$, est le sous-graphe de G induit par l'ensemble des sommets à distance au plus r de v .

Un graphe non-orienté est dit *connexe* lorsque entre toute paire de sommets, il existe un chemin commençant en l'un et terminant en l'autre. Un chemin de longueur au moins deux commençant et terminant par le même sommet est appelé un *cycle*. Une *corde* d'un cycle est une arête entre deux sommets non-consécutifs du cycle. Un graphe est une *forêt* s'il est sans cycle, et un *arbre* si c'est une forêt connexe. Un sommet de degré un d'un arbre est appelé une *feuille*, les autres sommets sont dits internes. Un arbre est dit *binnaire* lorsque tous ses sommets internes sont de degré trois. Un arbre à n sommets possède exactement $n - 1$ arêtes. Un ensemble de sommets $V' \in V(G)$ est dit *séparateur* lorsque $G - V'$ n'est pas connexe.

Un graphe non-orienté est dit *k -connexe* lorsqu'il faut supprimer au moins k sommets du graphe pour le déconnecter. De manière équivalente, un graphe est *k -connexe* lorsqu'il existe, entre toute paire de sommets du graphe, au moins k chemins sommet-disjoints (hormis le premier et le dernier sommet) reliant ces deux sommets et que k est minimum avec cette propriété. En particulier, un graphe est *2-connexe* ou encore *biconnexe* lorsque toute paire de sommets appartient à au moins un cycle du graphe.

Une *composante connexe* de G est un sous-graphe connexe maximal (au sens de l'inclusion) de G . La *connectivité* d'un sommet $v \in V(G)$ est le nombre de composantes connexes de $G - v$ moins le nombre de composantes connexes de G , i.e. c'est le nombre de nouvelles composantes connexes créées par la suppression du sommet v . Un *sommet d'articulation* du graphe G est un sommet $v \in V(G)$ ayant une connectivité strictement positive. Un *pont* de G est une arête ayant un sommet d'articulation de G à chacune de ses extrémités.

Un *isomorphisme* entre les graphes G et G' est une bijection $\phi : V(G) \rightarrow V(G')$ telle que pour tout $v, v' \in V(G)$, $\{v, v'\} \in E(G) \iff \{\phi(v), \phi(v')\} \in E(G')$. Une *contraction d'arête* de G est une opération supprimant une arête de G tout en fusionnant les deux sommets qui lui sont adjacents. Un graphe H est dit *mineur* de G s'il est isomorphe à un graphe obtenu à partir de G par une suite de contractions et de suppressions d'arêtes ou de suppressions de sommets isolés de G . Une classe de graphe est dite *stable* par mineurs si tout mineur de tout graphe appartenant à la classe appartient aussi à la classe.

Quelques graphes particuliers

Étant donné deux graphes G et H . Nous définissons l'*union jointe* de G et de H comme le graphe, noté $G \cup_j H$ et tel que

$$(V(G) \cup V(H), E(G) \cup E(H) \cup \{\{x, y\} \mid x \in G, y \in H\}).$$

Et le produit cartésien de G et de H comme le graphe, noté $G \times H$ et tel que

$$(V(G) \times V(H), \{\{(x, x'), (y, y')\} \mid (x = y \wedge \{x', y'\} \in E(H)) \vee (x' = y' \wedge \{x, y\} \in E(G))\}).$$

La *grille* $n \times n$ est alors le produit cartésien de deux chemins de longueur n . Et le graphe étoilé la donné d'un graphe de n sommets dont un unique sommet est relié à tous les autres. Cet unique sommet sera dénoté par le *centre* du graphe étoilé.

Si il désire plus détails autour des différents concepts détaillés ici, le lecteur est renvoyé à la monographie de Diestel sur différents sujets généraux de théorie des graphes [Diestel, 2012].

Un peu de topologie

Une *topologie* τ sur un ensemble E est un ensemble de parties de E stable par union quelconque et contenant \emptyset et E . Un *espace topologique* est la donnée d'un couple (E, τ) tel que les éléments de τ sont appelés les *ouverts* de (E, τ) . L'*intérieur* d'une partie d'un espace topologique est la réunion de tous les ouverts qu'elle contient.

Un *fermé* d'un espace topologique est le complémentaire d'un ouvert. La *fermeture* d'une partie d'un espace topologique est l'intersection de tous les fermés contenant cette partie. La *frontière* d'une partie d'un espace topologique est sa fermeture privée de son intérieur.

Une fonction $f : (E, \tau) \rightarrow (F, \theta)$ entre deux espaces topologiques est dite *continue* lorsque l'image réciproque $f^{-1}(O)$ d'un ouvert de (F, θ) est un ouvert de (E, τ) .

Nous considérons toujours la *droite* \mathbb{R} et le *plan* \mathbb{R}^2 en tant qu'espaces topologiques munis de leur topologie naturelle.

Une *courbe plane* est l'image d'une fonction continue $f : [0, 1] \rightarrow \mathbb{R}^2$ injective sur $]0, 1[$. Une courbe plane est dite *fermée* lorsque $\psi(0) = \psi(1)$. La restriction d'une courbe plane à l'ensemble $]0, 1[$ s'appelle un *arc*. Les *extrémités* d'un arc sont les images de 0 et de 1 par ψ .

Un théorème important sur les courbes planes est dû à Jordan.

Théorème 1 (Le théorème de Jordan sur les courbes planes)

Soit C une courbe plane fermée. Alors son complémentaire $\mathbb{R}^2 \setminus C$ consiste en exactement deux composantes connexes de \mathbb{R}^2 . L'une d'entre elles est bornée et l'autre ne l'est pas. La courbe C est ainsi le bord de chacune des deux composantes.

Graphe plan

Définition 2 (*Graphe plan*)

Un graphe plan est la donnée d'un couple d'ensembles finis $(\mathcal{V}, \mathcal{E})$, où

- \mathcal{V} est un ensemble de points du plan, appelés sommets du graphe,
- \mathcal{E} est un ensemble d'arcs du plan, appelés arêtes du graphe, dont les extrémités sont des sommets de \mathcal{V} ,
- deux arêtes différentes ont des ensembles différents d'extrémités,
- l'arc correspondant à une arête ne contient aucun sommet ni point d'un autre arc.

Un graphe plan $(\mathcal{V}, \mathcal{E})$ définit un graphe G tel que $V(G) = \mathcal{V}$ et $E(G) = \mathcal{E}$. Dans ce mémoire, nous utiliserons sans distinction un graphe plan et son graphe G associé et nous considérerons un graphe plan comme l'union des points du plan correspondant à ses sommets et à ses arêtes.

Une *face* d'un graphe plan G est une composante connexe de $\mathbb{R}^2 \setminus G$. Un sommet est dit incident ou adjacent à une face s'il appartient à sa frontière. Le *degré de face* d'un graphe plan G est le nombre maximum de sommets de G incidents à une même face de G . De même, une arête de G est dite incidente à une face de G , lorsqu'elle est contenue dans la frontière de cette face.

Le *graphe dual* d'un graphe plan G est un graphe plan, noté G^* , dont l'ensemble des sommets est l'ensemble des faces de G et tel que deux sommets de G^* sont adjacents si les frontières de leurs faces correspondantes dans G partagent une même arête.

Le *graphe radial* d'un graphe plan G est un graphe plan, noté R_G , qui se construit comme suit. À chaque face f_i de G est associé un sommet x_i tel que :

- $V(R_G) = V(G) \cup \{x_i \mid f_i \text{ est une face de } G\}$,
- $E(R_G) = \bigcup_{f_i \text{ face de } G} E_{f_i}$, où $E_{f_i} = \{\{x_i, v\} \mid v \text{ est incident à la face } f_i\}$.

La *distance radiale* entre deux sommets d'un graphe plan est la distance entre ces deux mêmes sommets dans son graphe radial.

Le *graphe médial* d'un graphe plan G est un graphe plan, noté M_G , dont l'ensemble des sommets est l'ensemble des arêtes de G et tel que deux sommets de M_G sont adjacents si leurs arêtes respectives dans G sont incidentes à une même face. Notons que le dual du graphe radial est le graphe médial.

Un *plongement* d'un graphe G dans le plan est un isomorphisme entre G et un graphe plan. Si un tel plongement existe, le graphe G est dit *planaire*.

La *superposition plane* de deux graphes plans G et H est le graphe, noté $G + H$ et tel que $(V(G) \cup V(H), E)$, où $E = E(G) \cup E(H)$ tel que s'il existe $\{x, y\} \in E(G)$ et $h \in V(H)$ tels que h se trouve dans l'arc du plan défini par l'arête $\{x, y\}$, alors $E = (E \cup \{\{x, h\}, \{h, y\}\}) \setminus \{\{x, y\}\}$. Notons que cette opération n'est pas commutative.

Un théorème important sur les plongements d'un graphe planaire est dû à Whitney.

Théorème 2 (*Whitney, 1932*)

Deux plongements quelconques d'un graphe planaire 3-connexe sont équivalents.

Cette définition nous sera relativement utile par la suite.

Définition 3

Une carte planaire est un graphe plan connexe dans lequel nous autorisons les boucles et les arêtes multiples.

Enfin, un excellent livre sur le sujet des graphes plongés dans les surfaces est dû à Mohar et Thomassen [Mohar and Thomassen, 2001]

1.1.2 Complexité paramétrée et problèmes de complétion

Complexité d'un problème

Un *alphabet* est un ensemble fini Σ de symboles. Un *mot* sur l'alphabet Σ est une suite, finie ou non, d'éléments de Σ . Nous noterons Σ^* , l'ensemble des mots sur l'alphabet Σ . Un *langage* sur l'alphabet Σ est une collection $L \subseteq \Sigma^*$ de mots sur Σ . Une *instance* d'un problème est un mot $I \in \Sigma^*$. Nous dirons que le langage L est associé au problème Π_L si L est l'ensemble des instances positives de Π_L . Décider du problème Π_L consiste à fournir un algorithme, se terminant en un temps fini, prenant en entrée une instance $I \in \Sigma^*$ et décidant de l'appartenance de I au langage L .

Un problème appartient à la classe NON-DETERMINISTIC POLYNOMIAL (NP) s'il existe un algorithme non déterministe et permettant de décider du problème en un nombre d'étapes polynomial en la taille de la donnée. Un problème Π est dit *plus difficile* que le problème Π' de la classe NP lorsque nous pouvons transformer toute instance $I' \in \Pi'$ en une instance $I \in \Pi$ en un temps polynomial en $|I'|$ et tel que $I \in \Pi \iff I' \in \Pi'$.

Un problème est dit NP-DIFFICILE quand il est plus difficile que tous les problèmes de la classe NP. Et il appartient à la classe NP-COMPLET, lorsqu'il est dans NP et qu'il est NP-DIFFICILE. En particulier, un problème est NP-DIFFICILE si il est plus difficile qu'au moins un problème de la classe NP-COMPLET. Pour une étude très complète de la classe NP-Difficile, nous renvoyons les lecteurs vers le célèbre livre de Garey et Johnson [Garey and Johnson, 1979].

Bien qu'il n'existe à ce jour aucun algorithme polynomial en la taille de la donnée pour les problèmes de la classe NP-DIFFICILE, certains d'entre eux sont si important qu'une solution obtenue de manière efficace est nécessaire. Plusieurs méthodes classiques sont utilisées pour traiter les problèmes NP-Difficiles. Parmi elles sont les méthodes d'approximations, donnant des solutions approchées de l'optimale avec une garantie de performances. Pour des définitions formelles et de plus amples explications, les lecteurs sont invités à se référer au livre écrit par Vazirani [Vazirani, 2001].

Une autre méthode formalisée par Downey et Fellow [Downey and Fellows, 1999] est la conception d'algorithmes paramétrés. Sa justification tient dans le fait que de se restreindre à la seule taille d'une instance du problème pour mesurer sa difficulté apparaît comme un obstacle. En effet, l'étude de la complexité d'un problème par rapport à d'autres paramètres structurels que sa simple taille a souvent été fructueuse aussi bien en terme d'algorithmes efficaces que de résultats théoriques. C'est la complexité paramétrée.

Complexité paramétrée

Une *paramétrisation* de Σ est une fonction $\kappa : \Sigma^* \rightarrow \mathbb{N}$ calculable en temps polynomial. Un *problème paramétré* sur Σ est la donnée d'un couple (Q, κ) , où $Q \subset \Sigma^*$ et κ est une paramétrisation de Σ . Une instance du problème paramétré (Q, κ) sur Σ est un mot $I \in \Sigma^*$ de paramètre $\kappa(I)$. Le problème paramétré (Q, κ) sur Σ admet, pour chaque instance $I \in \Sigma^*$, un problème de décision associé : a-t-on $I \in Q$ sachant $\kappa(I)$?

Un algorithme A est dit *polynomial à paramètre fixé* (FPT pour *Fixed-Parameter Tractable*), relativement à une paramétrisation κ de Σ , s'il existe une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$ calculable et un polynôme $p \in \mathbb{N}_0[X]$ tels que pour toute instance $I \in \Sigma^*$, le temps d'exécution de $A(I)$ soit au plus $f(\kappa(I)) \cdot p(|I|)$. Un problème paramétré (Q, κ) sur Σ est dit FPT s'il existe un algorithme FPT relativement à κ décidant Q . Pour plus de détail concernant la complexité paramétrée, nous renvoyons le lecteur vers la monographie de Flum et Grohe sur le sujet [Flum and Grohe, 2006].

Problème de complétion

Dans ce mémoire, nous allons nous centrer sur un type particulier de problème de modification de graphe. Le domaine général des problèmes de modification ou de transformation de graphe est assez récent. Il a été étudié pour la première fois, explicitement sous cette forme, par Yannakakis en 1981 [Yannakakis, 1981], lorsque ce dernier cherchait à montrer qu'un problème équivalent à celui de la complétion en graphe cordal était NP-Complet. Un graphe est dit *cordal* lorsque tout cycle de longueur au moins quatre admet une corde. Le problème de modification associé ici, est un problème de *complétion*, et consiste à décider si en ajoutant un certain nombre d'arêtes à un graphe, ce dernier admet une certaine propriété, ici le fait d'être cordal.

Les problèmes de modification sont centraux dans la théorie des graphes dans le sens où de nombreux problèmes de graphes peuvent se voir comme des problèmes de modification. Une *modification* d'un graphe peut être l'ajout ou la suppression d'une arête ou d'un sommet, la contraction d'une arête ou n'importe quelle combinaison de ces opérations. Le *problème de modification* associé à la propriété \mathcal{P} est le suivant. Étant donné une propriété \mathcal{P} représentée par une classe $\mathcal{C}_{\mathcal{P}}$, de graphes satisfaisant \mathcal{P} , et un graphe G , existe-t-il une suite finie de modifications de G telles que le graphe résultant soit dans $\mathcal{C}_{\mathcal{P}}$? Lorsque nous nous intéressons au nombre minimum de modifications à effectuer pour qu'un graphe satisfasse la propriété \mathcal{P} , nous cherchons en fait la distance, en terme de modifications, de ce graphe à la classe de graphes $\mathcal{C}_{\mathcal{P}}$.

De par leur généralité, les problèmes de modification ont été largement étudiés sous différentes formes. En effet, cette vision permet d'unifier et d'abstraire de nombreux résultats sur les graphes. En terme de complexité par exemple, Cai [Cai, 1996] a montré qu'un problème de modification, à l'exception de la contraction, sur une classe de graphes héréditaire caractérisée par un ensemble fini d'exclusions de sous-graphes était toujours FPT. Une classe de graphes est dite *héréditaire* lorsque la ou les propriétés qui la définissent sont stables par sous-graphe induit.

Parmi les problèmes de modification, une catégorie va nous intéresser plus particulièrement : les *problèmes de complétion d'arêtes*.

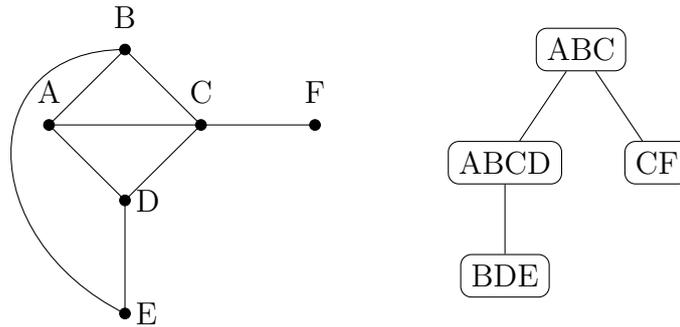


FIGURE 1.1 – Un graphe et sa décomposition arborescente.

 \mathcal{P} -COMPLÉTION

Entrée : G un graphe et \mathcal{P} une propriété sur les graphes.

Question : existe-t-il $F \subseteq (V(G) \times V(G)) \setminus E(G)$ tel que $G' = (V(G), E(G) \cup F)$ admette \mathcal{P} ?

Proposition 1

(Π) -Complétion $\in \mathcal{P} \implies \Pi \in \mathcal{P}$.

La preuve de cette proposition est immédiate si nous choisissons $F = \emptyset$. Voici quelques exemples intéressants :

- $\Pi =$ vérifier qu'un graphe est cordal ($\Pi \in \mathcal{P}$)
 $\rightarrow (\Pi)$ -Complétion \in NP-Complet [Yannakakis, 1981]
- $\Pi =$ vérifier qu'un graphe est hamiltonien ($\Pi \in$ NP-Complet)
 $\rightarrow (\Pi)$ -Complétion \in NP-Complet [Garey and Johnson, 1979]

Dans la suite, nous nous intéresserons à la complexité de deux problèmes de complétion dans les graphes plans. Dans ce cas, une classe associée à une propriété est une sous-classe de la classe des graphes plans. Ainsi, lors de l'ajout d'une arête à un graphe plan, nous devons nous assurer que le graphe résultant reste plan.

1.1.3 Décompositions de graphes**Largeur arborescente**

Certains problèmes, comme la couverture de sommets, admettent un algorithme polynomial lorsqu'ils sont restreint à des arbres. Pour généraliser cette idée, la largeur arborescente d'un graphe mesure sa ressemblance avec un arbre. En particulier, la largeur arborescente d'un arbre est de un. La notion de largeur arborescente a été introduite pour la première fois en 1986 par Robertson et Seymour [Robertson and Seymour, 1986] dans le cadre de leur preuve de la conjecture de Wagner. La largeur arborescente d'un graphe se définit par rapport à sa décomposition arborescente, voir Figure 1.1.

À chaque sommet t d'un arbre T est associé un ensemble de sommets $X_t \subseteq V(G)$ du graphe G appelé *sac*. Pour plus de clarté, nous appellerons *noeuds* les sommets de T , $N(T)$ l'ensemble des noeuds de T et $L(T)$ l'ensemble de ses feuilles. Une *décomposition arborescente* de G est la donnée d'un couple $(T, \{X_t\}_{t \in N(T)})$, où T est un arbre et pour tout $t \in N$, $X_t \subset V(G)$ tel que :

- pour tout $v \in V$, $\{t \in N : v \in X_t\}$ forme un sous-arbre de T ,
- pour tout $\{u, v\} \in E$, il existe $t \in N$ tel que $\{u, v\} \subseteq X_t$.

Remarquons qu'un sac associé à un noeud de la décomposition arborescente de G est toujours un ensemble de sommets séparateur de G .

La *largeur* d'une décomposition arborescente $(T, \{X_t\}_{t \in N})$ est alors

$$\max_{t \in N} |X_t| - 1.$$

La *largeur arborescente* d'un graphe G , notée $\mathbf{tw}(G)$, est donnée par le minimum de l'ensemble des largeurs des décompositions arborescentes de G . Une décomposition arborescente de G est dite optimale lorsque sa largeur est égale à la largeur arborescente de G .

Calcul de la largeur arborescente

Un graphe admet potentiellement un très grand nombre de décompositions arborescentes différentes. Ainsi, calculer la largeur arborescente d'un graphe ou encore trouver une décomposition arborescente optimale semble être des problèmes difficiles. En effet, Arnborg et al. ont montré en 1987 [Arnborg et al., 1987] que le calcul de la largeur arborescente d'un graphe donné était un problème NP-COMPLET. La proposition suivante résume les techniques de bases permettant de calculer la largeur arborescente d'un graphe donné.

Proposition 2 ([Bodlaender, 1998])

- Soit G un graphe dont les composantes connexes sont G_1, \dots, G_t , alors

$$\mathbf{tw}(G) = \max_{i \in [t]} \mathbf{tw}(G_i).$$

- Un cycle a une largeur arborescente de deux.
- Un graphe a une largeur arborescente de un si et seulement si c'est une forêt.
- $\mathbf{tw}(K_n) = n - 1$.
- La grille $n \times n$ a une largeur arborescente de n .
- Soient H et G deux graphes tels que $H \leq G$, alors $\mathbf{tw}(H) \leq \mathbf{tw}(G)$.
- Si pour $e \in E(G)$, nous avons le graphe $H = G - e$, alors

$$\mathbf{tw}(H) \leq \mathbf{tw}(G) \leq \mathbf{tw}(H) + 1.$$

- Si H est un mineur de G , alors $\mathbf{tw}(H) \leq \mathbf{tw}(G)$.

La proposition suivante nous sera très utile par la suite.

Proposition 3 ([Bodlaender, 1988], lemme 2.3)

Soient G un graphe et $k \in \mathbb{N}_{>0}$. Alors $\mathbf{tw}(G) \leq k$ si et seulement si $\mathbf{tw}(C) \leq k$, pour chaque composante 2-connexe C de G .

Enfin les trois propositions suivantes permettent de calculer les largeurs arborescentes d'objets radiales d'une manière relativement simple. Nous nous en servons aussi par la suite.

Proposition 4 (*[Koutsonas and Thilikos, 2011], lemme 4*)

Soient G un graphe plan et R_G son graphe radial. Alors $\mathbf{tw}(G + R_G) \leq 2 \cdot \mathbf{tw}(G)$.

Proposition 5 (*[Bodlaender, 1998]*)

Soient G un graphe plan et G^* son graphe dual. Alors $\mathbf{tw}(G^*) = \mathbf{tw}(G)$.

Et puisque le graphe médial est le dual du graphe radial, nous avons le corollaire suivant.

Corollaire 1

Soient G un graphe plan et M_G son graphe radial. Alors $\mathbf{tw}(G + M_G) \leq 2 \cdot \mathbf{tw}(G)$.

Trouver une décomposition arborescente

Un tour d'horizon assez récent concernant les différentes méthodes de calcul d'une décomposition arborescente optimale d'un graphe donné est proposé dans [Bodlaender, 2005]. Lors de ce mémoire, nous utiliserons cependant une 5-approximation pour obtenir une décomposition arborescente d'un graphe.

Cette méthode, exhibée dans [Bodlaender et al., 2013], consiste en un algorithme qui, étant donné un graphe G et un entier $q \in \mathbb{N}$, soit reporte que $\mathbf{tw}(G) > q$, soit construit une décomposition arborescente de G de largeur au plus $5 \cdot q$, et ce en $n \cdot 2^{O(q)}$ étapes.

Théorème 3 (*[Bodlaender et al., 2013]*)

Soient G un graphe et $q \in \mathbb{N}$. Alors il existe un algorithme qui, soit reporte que $\mathbf{tw}(G) > q$, soit construit une décomposition arborescente de G de largeur au plus $5 \cdot q$, et ce en $n \cdot 2^{O(q)}$ étapes.

Largeur de branche

La notion de largeur de branche d'un graphe a aussi été introduite par Robertson et Seymour dans [Robertson and Seymour, 1986]. C'est un concept très proche de la largeur arborescente mesurant de même la ressemblance d'un graphe à un arbre.

Une *décomposition en branches* de G est la donnée d'un couple (T, τ) telle que :

- T est un arbre binaire et
- $\tau : L(T) \rightarrow E(G)$ est une bijection.

Comme T est un arbre, la suppression de n'importe laquelle de ses arêtes crée deux composantes connexes, deux sous-arbres de T . Soient $e \in E(T)$ et T_1 et T_2 les deux composantes connexes de $T - e$, alors $L(T_1)$ et $L(T_2)$ définissent deux ensembles disjoints d'arêtes de G . Nous pouvons donc définir $G_1 = G[L(T_1)]$ et $G_2 = G[L(T_2)]$ les deux sous-graphes de G induits par les ensembles d'arêtes $L(T_1)$ et $L(T_2)$.

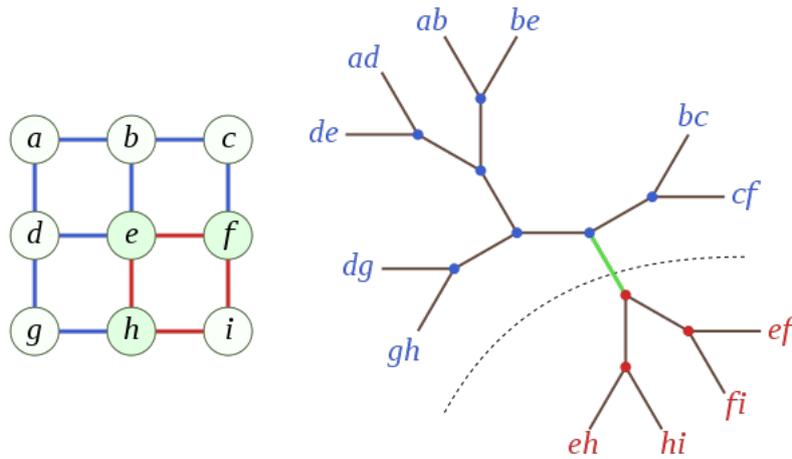


FIGURE 1.2 – Une décomposition arborescente optimale de la grille 3×3 , de largeur trois et séparée par l'arête verte ; son ensemble mitoyen est $\{e, f, h\}$.

L'ensemble mitoyen ("middle set") d'une arête $e \in E(T)$, noté $\mathbf{mid}(e)$, est $V(G_1) \cap V(G_2)$, i.e. l'ensemble des sommets de G à l'intersection des sous-graphes G_1 et G_2 de G . De même que les sacs d'une décomposition arborescente, les ensembles mitoyens d'une décomposition en branches de G sont des ensembles de sommets séparateurs de G . L'ordre d'une arête $e \in E(T)$ est $|\mathbf{mid}(e)|$, le cardinal de son ensemble mitoyen.

La largeur d'une décomposition en branches (T, τ) de G est le maximum des ordres des arêtes $e \in E(T)$. Et la largeur de branche de G , notée $\mathbf{bw}(G)$, est le minimum parmi les largeurs de branche de ses décompositions en branches. Une décomposition en branches de G est dite optimale lorsque sa largeur est égale à la largeur de branche de G . Pour une illustration, voir la Figure 1.2, tirée de [Eppstein, 2010].

Les deux propositions suivantes, caractérisant les liens entre la largeur de branche et la largeur arborescente d'un graphe, sont issues de l'article [Robertson and Seymour, 1991].

Proposition 6 ([Robertson and Seymour, 1991])

$$\mathbf{bw}(G) \leq \mathbf{tw}(G) + 1 \leq \lfloor \frac{3}{2} \cdot \mathbf{bw}(G) \rfloor.$$

Théorème 4 ([Robertson and Seymour, 1991])

Soit G un graphe plan et (T, τ) une décomposition arborescente de G de largeur $q \in \mathbb{N}$. Alors il existe un algorithme donnant une décomposition en branches de G de largeur q en $O(n^2)$ étapes.

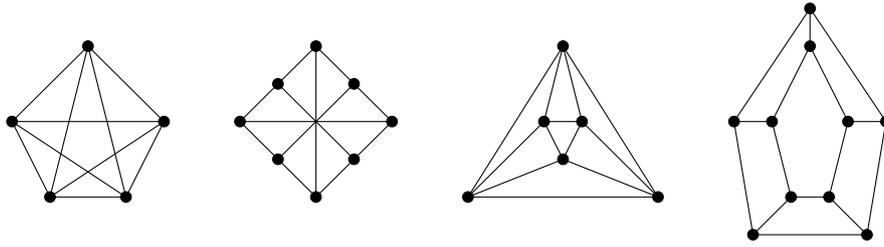


FIGURE 1.3 – Les mineurs exclus pour les graphes à largeur arborescente au plus trois.

1.2 Largeurs bornées et algorithmes FPT

1.2.1 Mineurs

Dans une série de papiers parus entre 1983 et 2004, Robertson et Seymour ont prouvé la conjecture de Wagner, stipulant que la relation de mineur entre deux graphes est un bel ordre sur l'ensemble des graphes non-orientés, finis. Ainsi, toute classe de graphes stable par mineurs admet un ensemble fini de mineurs exclus, i.e. mineurs minimaux n'appartenant pas à la classe. Ce théorème permet de caractériser une classe de graphes d'un point de vue formel, en donnant son ensemble fini de mineurs exclus.

Par exemple, la classe des graphes *planaires* est caractérisée par l'exclusion des mineurs K_5 et $K_{3,3}$, c'est le théorème de Wagner. Ou encore, il est possible de caractériser la classe des graphes dont la largeur arborescente est plus petite que $k \in \mathbb{N}$. Arnborg et al. [Arnborg et al., 1990] en ont donné une caractérisation pour les largeurs arborescentes inférieures à trois :

- la largeur arborescente d'un graphe est d'au plus un si et seulement si il ne contient pas K_3 comme mineur,
- elle est d'au plus deux si et seulement si il ne contient pas K_4 comme mineur,
- et d'au plus trois si et seulement si il ne contient aucun des quatre graphes de la Figure 1.3 comme mineurs.

Une autre conséquence du théorème de Robertson et Seymour est le fait que dans toute classe de graphes stable par mineurs, il existe un algorithme pouvant décider de l'appartenance d'un graphe à cette classe.

En effet, Robertson et Seymour [Robertson and Seymour, 1991] donnent un algorithme FPT décidant si un graphe H est mineur de G en $f(|H|) \cdot O(n^3)$ étapes, où f est une fonction gigantesque en terme de complexité. Ainsi, quand l'ensemble fini des mineurs exclus d'une classe de graphes stable par mineurs est entièrement connue, il est possible de savoir si un graphe donné appartient ou non à la classe en testant sur ce graphe la relation de mineur, et ce sur chaque mineur exclus.

Mais l'ensemble fini des mineurs exclus n'est en général pas connu. De plus, la fonction f précédente est gigantesque en terme de complexité. Cet algorithme n'est donc pas réalisable en pratique.

Cependant, plusieurs méthodes pour créer des algorithmes efficaces lorsque la largeur arborescente est bornée existent. Parmi celles-ci, la principale en est la programmation dynamique sur les graphes. Le paragraphe suivant est issu de la présentation pédagogique qu'en fait Bodlaender dans [Bodlaender, 1997].

1.2.2 Programmation dynamique

Largeur arborescente bornée

Résoudre un problème sur une classe de graphes à largeur arborescente bornée par une programmation dynamique a été expliqué pour la première fois dans [Arnborg et al., 1987] et fonctionne comme suit. Soit une décomposition arborescente (T, \mathcal{X}) de G à largeur bornée. Il faut alors subdiviser une arête de T choisie arbitrairement, en créant un nouveau noeud r que nous définirons comme la racine de T . Le sac correspondant au noeud r est défini comme l'ensemble vide de sommets de G . Puis orienter les arêtes de T de telle sorte qu'il existe un unique chemin orienté vers r depuis tout autre noeud de l'arbre. Notons qu'un noeud est soit une feuille, soit a au moins un parent.

À chaque noeud de T est alors associé une solution partielle du problème. La solution associée à la racine étant la solution finale. Chaque solution partielle est stockée dans une table dont la taille dépend de la taille du sac de son noeud associé, i.e. de la largeur arborescente de la décomposition. Nous calculons d'abord les solutions partielles associées à chaque feuille de l'arbre puis nous parcourons l'arbre du bas vers le haut, i.e. en remontant vers la racine r . La solution partielle de chaque noeud est alors calculée grâce aux solutions partielles de ses noeuds parents.

Comme la taille des tables ne dépend que de la largeur arborescente de la décomposition, l'algorithme obtenu par cette programmation dynamique est un algorithme FPT paramétrisé par la largeur arborescente de la décomposition.

Largeur arborescente localement bornée

Comme la grille $n \times n$ est un graphe planaire et que sa largeur arborescente est de n , la classe des graphes planaires n'admet pas de largeur arborescente bornée. La notion de largeur arborescente localement bornée, développée ici, permet quand même d'obtenir des résultats intéressants d'un point de vue algorithmique sur ce genre de classe de graphes.

Une méthode, créée par Baker alors qu'elle cherchait des algorithmes d'approximation pour des problèmes sur les graphes planaires [Baker, 1994] et formalisée par Eppstein [Eppstein, 2000], est à l'origine du concept de largeur arborescente localement bornée.

Nous avons tout d'abord le théorème suivant.

Théorème 5 ([Eppstein, 1995])

Dans un graphe planaire, la largeur arborescente est bornée par une fonction linéaire du diamètre du graphe.

Si le diamètre n'est pas borné, la technique de Baker consiste à recouvrir le graphe avec des r -voisinages dont le diamètre est par construction borné et donc avec des sous-graphes induits à largeur arborescente bornée. Et si dans un graphe, une telle construction est possible, le graphe est alors dit à largeur arborescente localement bornée, et qui ne dépend localement que de r . Notons que tout graphe planaire n'est pas à diamètre borné puisque la grille $n \times n$ a un diamètre de n^2 . Mais le théorème précédent implique cependant qu'un graphe planaire a une largeur arborescente localement bornée.

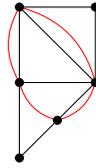


FIGURE 1.4 – Lasso de taille 4

En fait, dans [Eppstein, 2000], Eppstein donne une caractérisation des graphes à largeur arborescente localement bornée. La caractérisation s’effectue via les graphes apex. Un graphe est dit *apex* si la suppression d’un de ses sommets le rend planaire. Un exemple de graphe apex est la grille $n \times n$ à laquelle est ajouté un sommet relié à tous les autres. Eppstein énonce alors qu’une classe de graphes fermées par mineurs a une largeur arborescente localement bornée si et seulement si elle ne contient aucun graphe apex.

La technique de Baker a été au départ utilisée pour construire des algorithmes d’approximation efficaces sur les graphes planaires. En effet, cette technique décompose un graphe planaire plongé dans le plan en épluchant successivement la face externe et attribue à chaque sommet un entier correspondant à la couche sur laquelle il se trouvait lorsqu’il a été épluché. Si par cet épluchage, le graphe vide est atteint après k suppressions de couches, le graphe est dit k -planair extérieur.

Beaucoup de problèmes admettent une résolution polynomiale quand ils sont restreints aux graphes planaires extérieurs. La technique de Baker consiste alors à résoudre les problèmes de façon optimale sur chaque couche, qui est un graphe planair extérieur, puis à les recombinaer en ne choisissant que les meilleures solutions parmi chaque couche. Baker [Baker, 1994] montre alors que cette méthode donne une solution à un facteur $(1 + \frac{1}{k})$ de l’optimale.

Enfin, Eppstein [Eppstein, 2000] a montré que cette technique s’appliquait aussi sur les graphes dont la largeur arborescente est localement bornée, et ce de la même manière que sur les graphes k -planaires extérieurs. Chaque couche obtenue admet alors une largeur arborescente bornée et il devient possible de résoudre le problème via une programmation dynamique sur chacune d’entre elles. Le problème d’isomorphisme de sous-graphe a été prouvé FPT, paramétré par la taille du sous-graphe, par Eppstein de cette manière là [Eppstein, 1995].

Décomposition en coupes sphériques

Étant donné un graphe plan G et une décomposition en branches (T, μ) de G à largeur de branche bornée, nous pouvons de même définir une programmation dynamique impliquant un algorithme FPT paramétré par la largeur de branche de la décomposition.

Une courbe plane fermée O est appelée un *lasso* de G si elle intersecte G uniquement en une partie de ses sommets. Nous posons $V_O = V(G) \cap O$, voir Figure 1.4. Un *arc* d’un lasso est une composante connexe de $O \setminus V_O$, et par convention, si $V_O = \emptyset$, O ne possède pas d’arc.

Définition 4 (Décomposition en coupes sphériques)

Une décomposition en coupes sphériques de G est la donnée d'un triplet (T, μ, π) , où (T, μ) est une décomposition en branches de G et π est une fonction envoyant chaque $e \in E(T)$ sur un ordre cyclique des sommets de G , tel que pour chaque $e \in E(T)$, il existe un lasso O_e satisfaisant les propriétés suivantes :

- O_e intersecte chaque face de G au plus une fois,
- G_1^e est contenu dans un des disques fermés et bornés par O_e et G_2^e est contenu dans l'autre,
- $\pi(e)$ est un ordre cyclique de V_{O_e} défini par un parcours de O_e dans le sens des aiguilles d'une montre.

La largeur d'une décomposition en coupes sphérique est la largeur de la décomposition en branches à partir de laquelle elle est définie.

Le théorème suivant relie la décomposition en coupes sphériques et la décomposition en branches d'un graphe 2-connexe.

Théorème 6 ([Seymour and Thomas, 1994], (5.1))

Soit G un graphe 2-connexe, avec au moins deux sommets, et une décomposition en branches de largeur au plus $k \in \mathbb{N}$. Alors nous pouvons construire une décomposition en coupes sphériques de G de largeur au plus k en $O(n^2)$ étapes.

2.1 Problèmes et résultats

2.1.1 Définitions

Dans un papier de 1987 [Chung, 1987] sur les problèmes relatifs au diamètre dans les graphes planaires, Chung a introduit le problème suivant : peut-on ajouter des arêtes à un graphe plan G de sorte à ce que le graphe résultant reste plan et admette un diamètre d'au plus $d \in \mathbb{N}$?

Définition 5 (*Complétion plane d'un graphe*)

Soient G un graphe plan et $F \in (V(G) \times V(G)) \setminus E(G)$. Alors le graphe $(V(G), E(G) \cup F)$, noté G^+ est appelé complétion plane de G si il est plan.

COMPLÉTION PLANE À DIAMÈTRE BORNÉ (CPDB)

Entrée : un graphe plan G et $d \in \mathbb{N}$.

Question : existe-t-il une complétion plane G^+ de G admettant un diamètre d'au plus d ?

Ce problème est donc un problème de complétion d'arêtes, une illustration en est donnée par la Figure 2.1.

2.1.2 Complexités

Le problème de la COMPLÉTION PLANE À DIAMÈTRE BORNÉ a été montré NP-COMPLET dans le cas orienté par Fellows et Diejter en 1993 dans un article jamais soumis à la publication [Dejter and Fellows, 1993]. Ils montrent ce résultat via une réduction depuis le problème PLANAR 3-SAT.

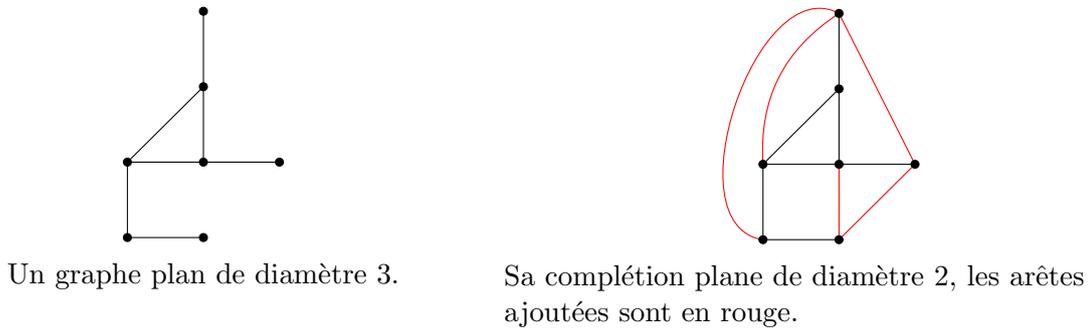


FIGURE 2.1

Théorème 7 (*[Dejter and Fellows, 1993]*)

Le problème de la COMPLÉTION PLANE À DIAMÈTRE BORNÉ est NP-COMPLET dans le cas orienté.

Ils ont de plus, et dans ce même papier, donné une démonstration non-constructive de l'existence d'un algorithme FPT, paramétrisé par d , permettant de résoudre ce problème.

Théorème 8 (*[Dejter and Fellows, 1993]*)

Le problème de la COMPLÉTION PLANE À DIAMÈTRE BORNÉ est FPT paramétré par d .

Démonstration : Tout d'abord, notons que l'ensemble des instances positives (G, d) du problème est stable sous l'opération de mineurs de graphe. En effet, un mineur de G est aussi une instance positive (avec le même d) puisque, soit des arêtes ont été supprimées pour obtenir le mineur et dans ce cas elles peuvent être ajoutées par complétion, soit des arêtes ont été contractées et dans ce cas le diamètre général du graphe a été réduit.

Le célèbre théorème de Robertson et Seymour nous permet alors de dire que pour un d fixé, la famille des instances positives (G, d) peut être caractérisée par un ensemble finie de mineurs interdits. De plus, Robertson et Seymour nous donne un algorithme permettant de vérifier si un graphe donné possède un graphe H en tant que mineur. Cet algorithme est FPT paramétré par $|H|$. ■

Mais même pour de petites valeurs de d , l'ensemble des mineurs interdits nous est encore inconnu. De plus, l'algorithme permettant de vérifier si un graphe est mineur d'un autre est certes FPT, mais la constante cachée à l'intérieur de sa complexité est gigantesque. Ainsi, cette preuve ne donne en aucun cas une méthode pour construire un algorithme. Fellows et Dejter ont alors laissé le problème ouvert suivant.

Problème ouvert : trouver un algorithme FPT, paramétré par d , et réalisable pour le problème de la COMPLÉTION PLANE À DIAMÈTRE BORNÉ dans les cas orienté et non-orienté.

Dans ce mémoire, nous nous intéresserons seulement au cas non-orienté.

2.1.3 Variantes

Pour nous attaquer au problème ouvert précédent, nous proposons d'essayer de comprendre la complexité du problème de la COMPLÉTION PLANE À DIAMÈTRE BORNÉ lorsque nous ajoutons d'autres paramètres que d . Un paramètre non-structurel mais intéressant, puisque dépendant du problème même, est celui du nombre d'arêtes ajoutées. En effet, il est naturel de se demander quel est le nombre minimum d'arêtes à ajouter afin que le diamètre devienne inférieur à d . Ce problème de minimisation admet un problème de décision associé auquel nous allons donc nous intéresser.

Nous proposons ainsi trois variantes du problème de la COMPLÉTION PLANE À DIAMÈTRE BORNÉ. Trois variantes autour du paramètre supplémentaire $k \in \mathbb{N}$ représentant le nombre d'arêtes ajoutées lors de la complétion, i.e. autour du problème de décision associé au problème de minimisation.

BUDGET-COMPLÉTION PLANE À DIAMÈTRE BORNÉ

Entrée : un graphe plan G et $d, k \in \mathbb{N}$.

Question : existe-t-il $F \in (V(G) \times V(G)) \setminus E(G)$ tel que $|F| \leq k$ et la complétion plane $(V(G), E(G) \cup F)$ admette un diamètre d'au plus d ?

Cette variante rajoute la contrainte que le nombre total d'arêtes ajoutées lors d'une complétion plane de G doit être d'au plus k . Remarquons que puisque la planarité doit être conservée, tout arête ajoutée lors d'une complétion doit l'être à l'intérieur d'une face de G . De plus, le nombre maximum d'arêtes qu'il est possible d'ajouter à l'intérieur d'une face incidente à k sommets est de $k - 2$. Cela correspond au nombre maximum de couplages non-croisés d'un cycle.

Proposition 7

Le nombre maximum de couplages non-croisés d'un cycle de taille $k \in \mathbb{N}$ est de $k - 2$.

Cette proposition a inspirée la variante suivante.

FACES BORNÉES-COMPLÉTION PLANE À DIAMÈTRE BORNÉ

Entrée : un graphe plan G et $d, b \in \mathbb{N}$ tels que chaque face soit incidente à au plus b sommets de G .

Question : existe-t-il une complétion plane G^+ de G admettant un diamètre d'au plus d ?

Enfin, la dernière variante généralise le problème FACES BORNÉES-COMPLÉTION PLANE À DIAMÈTRE BORNÉ. En effet, ici nous allons borner le nombre d'arêtes ajoutées par face de G .

BUDGET PAR FACE-COMPLÉTION PLANE À DIAMÈTRE BORNÉ

Entrée : un graphe plan G et $d, k' \in \mathbb{N}$ tels que chaque face soit incidente à au plus k' sommets de G .

Question : existe-t-il une complétion plane G^+ de G admettant un diamètre d'au plus d ?

Cette variante est plus générale que la première variante puisqu'il nous suffit de prendre $k = k' \cdot f$, où f dénote le nombre de faces de G , pour obtenir une instance du problème de la BUDGET-COMPLÉTION PLANE À DIAMÈTRE BORNÉ. De même, cette variante généralise aussi la deuxième variante, puisqu'il nous suffit de prendre $b = k'$ pour obtenir une instance de FACES BORNÉES-COMPLÉTION PLANE À DIAMÈTRE BORNÉ.

Dans la suite et pour plus de clarté, nous dénoterons par

- BCP le problème de la BUDGET-COMPLÉTION PLANE À DIAMÈTRE BORNÉ,
- FBC le problème de la FACES BORNÉE-COMPLÉTION EN DIAMÈTRE BORNÉ et par
- BFCP celui de la BUDGET PAR FACE-COMPLÉTION PLANE À DIAMÈTRE BORNÉ.

Nous avons alors le résultat suivant, donné par Golovach, mais qui n'existe actuellement que sous une forme manuscrite.

Théorème 9 (Golovach, 2014)

Le problème BCP est NP-Complet.

La réduction polynomiale est effectuée à partir du problème PLANAR 3-SAT. Notons le corollaire important de ce théorème.

Corollaire 2

Le problème de la COMPLÉTION PLANE À DIAMÈTRE BORNÉ est NP-COMPLET dans le cas non-orienté.

En effet, il nous suffit de poser que $k = \frac{n(n-1)}{2}$.

2.1.4 Un algorithme FPT paramétré par d et k

Dans ce mémoire, nous décrivons un algorithme FPT, paramétré par d et k , pour résoudre le problème BCP. Cet algorithme est basé sur une programmation dynamique.

Cependant, nous avons réussi à décrire la programmation dynamique pour le problème BFCP. Mais dans ce cas, l'algorithme précédent n'est plus FPT paramétré par d et k .

Pour ce faire, nous allons montrer que le problème de la BFCP est équivalent au problème de la (d, k, s, w) -arête colorabilité sur une instance transformée. Puis nous donnerons une programmation dynamique sur ce problème équivalent, pour enfin exhiber l'algorithme général.

Et puisque le problème BFCP est plus général que le problème BCP, nous allons pouvoir sans peine adapter l'algorithme à ce dernier. Ce qui nous donnera le théorème suivant.

Théorème 10

Étant donné un graphe plan G_{input} tel que $|V(G_{\text{input}})| = n \in \mathbb{N}$ et deux entiers $d, k \in \mathbb{N}$. Alors il existe un algorithme FPT, paramétré par d et k , pouvant décider si (G_{input}, d, k) est une instance positive du problème BCP.

2.2 Augmentation radiale

2.2.1 Grille annulaire

Parce qu'il faut préserver la planarité de G lors d'une complétion, toute arête est forcément ajoutée dans une face déjà existante. La potentialité pour une face F de G de se voir ajouter des arêtes peut être simulée via un gadget. Le graphe \overline{F} composé de F et de ses arêtes ajoutées sera alors un mineur de ce gadget. Encore mieux : quelque soit la façon d'ajouter des arêtes dans F , tant qu'aucun croisement n'est créé et que leur nombre ne dépasse pas k , il existe une suite de contractions et de suppressions d'arêtes du gadget permettant d'obtenir \overline{F} .

Ce gadget s'appelle une grille annulaire et est défini comme suit. La $(k \times r)$ -grille annulaire, notée $\Gamma_{k,r}$, est le produit cartésien d'un chemin de $k + 1$ sommets et d'un cycle de r sommets, où k et r sont des entiers positifs tels que $k \geq 2$ et $r \geq 3$. La proposition suivante est donnée par Diestel dans [Diestel, 2012].

Proposition 8 ([Diestel, 2012])

Dans un graphe plan 2-connexe, la frontière de toute face est composée d'un cycle unique.

Si $r \neq 4$, alors $\Gamma_{k,r}$ admet exactement deux faces f_1 et f_2 dont la frontière ne forme pas un carré. Si $r = 4$, nous choisissons f_1 et f_2 comme deux faces dont tous les sommets incidents sont de degré trois et telles que leurs frontières n'ont aucun sommet en commun. Nous appellerons le cycle de la frontière de f_1 la *base* de $\Gamma_{k,r}$ et le cycle de la frontière de f_2 le *toit* de $\Gamma_{k,r}$, voir Figure 2.2. Si F est une face de G dont la frontière est composée d'un seul cycle, nous pouvons alors, dans G , ajouter $\Gamma_{k,r}$ sur F en associant le cycle de la frontière de F avec le toit de $\Gamma_{k,r}$.

La définition suivante va nous permettre de prouver la simulation via un algorithme qui agira couche par couche sur la grille annulaire.

Définition 6 (*ℓ -couche*)

Soient $\ell \in [k]$. La ℓ -couche de $\Gamma_{k,r}$ est la ligne induite par le produit cartésien des ℓ et $(\ell + 1)$ -ième sommets du chemin de taille k avec le cycle de r sommets dans $\Gamma_{k,r}$, duquel nous retirons le cycle bornant sa face extérieure.

Cette définition nous sera utile pour calculer la largeur arborescente d'une grille annulaire.

Définition 7 (*Ligne*)

À partir d'un sommet x de la grille annulaire $\Gamma_{k,r}$, nous définissons sa ligne, notée $L_x = \{x_1, \dots, x_k\}$, comme l'ensemble des sommets de la grille annulaire composant un chemin de taille $k - 1$ partant de x et passant une fois exactement sur chaque couche. Et posons que x_ℓ sera le sommet de la ℓ -couche de la ligne L_x , pour chaque $\ell \in [k]$.

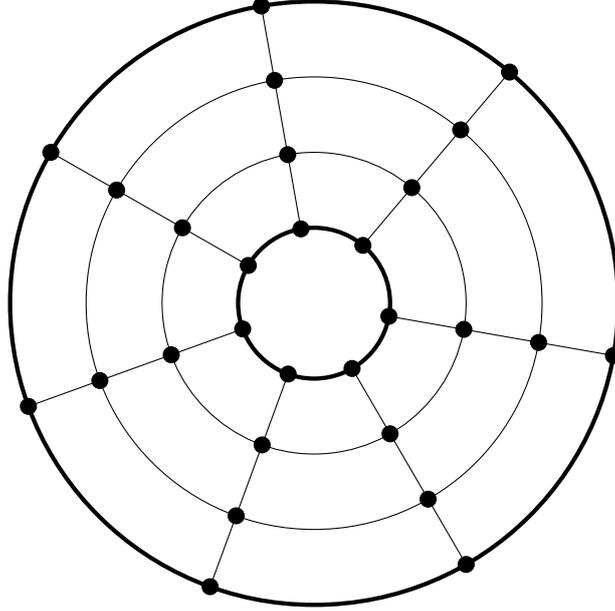


FIGURE 2.2 – Le graphe $\Gamma_{7,4}$ dont la base et le toit sont respectivement les cercles en gras de plus petit et de plus grand rayon.

Enfin, cette proposition justifie le fait que nous puissions plonger une définition abstraite de la grille annulaire d'une manière unique dans le plan.

Proposition 9

$\Gamma_{k,r}$ est un graphe planaire 3-connexe et admet ainsi un plongement unique, à équivalence près, dans le plan.

Démonstration : Nous allons montrer la 3-connexité de $\Gamma_{k,r}$ par le fait qu'il existe trois chemins sommet-disjoints (hormis leurs points de départ et d'arrivée) entre toute paire de points de $\Gamma_{k,r}$. Soient alors x et $y \in V(\Gamma_{k,r})$ tels que x est sur la t -couche et y sur la s -couche de $\Gamma_{k,r}$, avec $s \leq t \in \mathbb{N}$.

Le premier chemin sera la concaténation du plus court chemin de x à x_s et du plus court chemin de x_s à y . Le second chemin quant à lui sera la concaténation du plus court chemin de x à y_t et du plus court chemin de y_t à y . Enfin, si le plus court chemin de x à y_t va à partir de x dans le sens des aiguilles d'une montre (resp. trigonométrique), alors le troisième chemin sera la concaténation du plus court chemin de x au dernier sommet z de la t -couche rencontré avant y_t dans le sens trigonométrique (resp. des aiguilles d'une montre), du plus court chemin de z à z_s et du plus court chemin de z_s à y_s . ■

Puisque le problème BFCP nous autorise à ajouter au plus k arêtes dans chaque face, une grille annulaire $\Gamma_{k,r}$ sera ajoutée sur chacune des faces de G . Cependant, pour que cela devienne possible, il faut que la frontière de chaque face soit réduite à un cycle, ou comme nous le verrons plus tard, à une union de cycles.

Ainsi, dans la sous-section suivante, nous allons présenter une méthode permettant de transformer un graphe plan de manière à ce que la frontière de chacune de ses faces soit une union de cycles, i.e. devienne lisse.

2.2.2 Graphe lisse

Définition 8

Un graphe est lisse lorsque chacune de ses composantes est 2-connecte, i.e. ne possède aucun sommet d'articulation.

La conséquence suivante justifie l'utilisation des graphes lisses.

Proposition 10

La frontière de chacune des faces d'un graphe lisse est une union de cycles.

Démonstration : Supposons que G soit un graphe lisse, i.e. tel que toutes ses composantes soient 2-connectes. Soient alors F une face de G et C_1, \dots, C_t les composantes connexes de G incluses dans l'intérieur de F . Considérons alors C_i pour $i \in [t]$ fixé. Par hypothèse, le sous graphe de G induit par C est un graphe 2-connecte. Et par la Proposition 8, la frontière de chacune de ses faces est réduite à un cycle. En particulier, la restriction de la frontière de F à C_i est un cycle. Ainsi, la frontière de F est l'union des cycles de chaque C_i , pour $i \in [t]$. ■

Un algorithme

L'algorithme permettant de transformer un graphe plan en un graphe lisse fonctionne comme suit. À chaque sommet d'articulation $x \in V(G)$ de G , est appliquée la procédure ci-dessous, illustrée par la Figure 2.3. Rappelons que $p(x)$ est la connectivité du sommet d'articulation x .

La procédure **grapheLisse** permettant de transformer un graphe plan en un graphe lisse est décrite dans l'Algorithme 1. Cet algorithme utilise la procédure **sommetLisse** permettant de lisser un sommet d'articulation de G . Cette dernière procédure est décrite par l'Algorithme 2.

Entrées : Un graphe plan G et un sommet d'articulation x de G .

Output : Un graphe lisse G' et $J \subseteq E(G')$.

début

$G' = G$
$J, J' = \emptyset$
tant que $\exists x \in V(G) : x$ est un sommet d'articulation de G faire
$(G', J') = \text{sommetLisse}(G, x)$
$J = J \cup J'$
fin
retourner (G', J)
fin

Algorithme 1: grapheLisse

Entrées : Un graphe plan G et un sommet d'articulation x de G .

Output : Un graphe plan G' ayant un sommet d'articulation de moins que G et $J \subseteq E(G')$.

début

$G' = G$

Soient $C_1, \dots, C_{p(x)}$ les composantes connexes de $G - x$ qui ne sont pas des composantes connexes dans G

pour $i \in [p(x)]$ **faire**

 effectuer un parcours de C_i dans G , partant de x et revenant à x

 soient a_{i_1} et a_{i_2} les premier et dernier sommets différents de x de ce parcours

 stocker tous les sommets de $C_i \setminus \{a_{i_1}, a_{i_2}\}$ incidents à x dans un ensemble S_i

fin

$V(G') = (V(G') \cup \{x_1, \dots, x_p\}) \setminus \{x\}$

$E(G') = (E(G')) \cup \{\{x_i, x_{i+1}\} \mid i \in [p]\}$ (avec $x_{p+1} = x_1$)

$J = \{x_1, \dots, x_p\}$

pour $i \in [p(x)]$ **faire**

$E(G') = E(G') \cup \{a_{i_1}, x_i\}, \{a_{i_2}, x_{i+1}\}$

pour $s \in S_i$ **faire**

$E(G') = E(G') \cup \{s, x_i\}$

fin

fin

retourner (G', J)

fin

Algorithme 2: sommetLisse

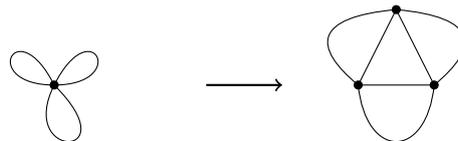


FIGURE 2.3 – Un sommet d'articulation de connectivité trois remplacé par un cycle.

Conséquences

Soient G un graphe plan, G' le graphe obtenu par l'Algorithme 1 appliqué à G , et J l'ensemble des arêtes des cycles remplaçant les sommets d'articulation de G dans G' .

Nous avons alors les résultats suivants.

Lemme 1

1. G' est lisse.
2. Le nombre de composantes connexes de G' est exactement le nombre de composantes connexes de G .
3. G est isomorphe au graphe obtenu à partir de G' en contractant les arêtes de J .
4. La procédure **grapheLisse** décrite par l'Algorithme 1 s'effectue en $O(n)$ étapes.
5. $|V(G')| \leq 2 \cdot |V(G)|$.
6. $\mathbf{tw}(G') \leq \mathbf{tw}(G) + 1$.

Démonstration: 1. À la fin de l'exécution de la procédure précédente pour un sommet d'articulation, aucun nouveau sommet d'articulation n'est créé et le nombre de sommets d'articulation diminue de un dans le graphe résultant. Ainsi, G' , le graphe obtenu à la fin de l'algorithme, est bien lisse puisqu'il ne possède plus aucun sommet d'articulation.

2. De même, à la fin de chaque itération de la procédure, aucune nouvelle composante connexe n'est créée. Donc le nombre de composante de G' est donc égal à celui de G .

3. Le graphe résultant lorsque nous contractons toutes les arêtes d'un cycle est le graphe réduit à un unique sommet. Ainsi, lorsque nous contractons les arêtes de J dans G' , nous identifions chaque cycle créé par l'algorithme à un sommet unique, et toutes les arêtes dont exactement une extrémité est un sommet du cycle sont maintenant adjacentes à cet unique sommet. Le graphe résultant est alors isomorphe à G .

4. La complexité de cet algorithme dépend du nombre de fois où la procédure **sommetLisse** décrite par l'Algorithme 2 est exécutée, et par la complexité de cette dernière. La procédure **sommetLisse** effectue un parcours de graphe de la taille des nouvelles composantes connexes de $G - x$. Cette procédure est appelée autant de fois qu'il y a de sommets d'articulation dans G .

Cependant une analyse amortie de la complexité nous montre que le nombre de sommets d'articulation et la taille des nouvelles composantes connexes sont complémentaires, i.e. nous ne pouvons avoir à la fois beaucoup de sommets d'articulation et de grandes composantes connexes autour de ces sommets dans un graphe. La complexité est donc bien en $O(n)$.

5. Le nombre de sommets ajoutés dans G' dépend du nombre de cycles ou de chemins remplaçant les sommets d'articulation et de leurs tailles. Du plus, la taille du cycle ou du chemin remplaçant un sommet d'articulation donné dépend de la connectivité de ce sommet.

Remarquons de plus que supprimer une arête quelconque de n'importe quel cycle de G ne diminue la connectivité d'aucun sommet d'articulation de G ni leur nombre. Ainsi, l'Algorithme 1 appliqué sur G ou sur un G modifié dans lequel nous aurions enlevé une arête de chaque cycle, i.e. un graphe sans cycle, retourne un graphe qui a exactement

le même nombre de sommets. Nous pouvons donc compter les sommets ajoutés dans G' comme dans G'' , le résultat de l'Algorithme 1 appliqué au G modifié sans cycle.

Pour compter les sommets ajoutés de G'' , il suffit de compter les sommets ajoutés de chaque composante connexe C' de G'' . Soit C la composante connexe de G correspondant à C' . Remarquons que C' est un arbre et que dans un arbre tout sommet interne, i.e. qui n'est pas une feuille, est un sommet d'articulation et sa connectivité correspond exactement à son degré. Le nombre de sommets ajoutés dans C'' est donc borné par la somme des degrés de ses sommets, i.e.

$$|V(C') - V(C)| \leq 2 \cdot \sum_{x \in C'} \delta(x) = 2 \cdot (|V(C')| - 1) = O(|V(C)|).$$

6. Il nous reste à montrer que $\mathbf{tw}(G') \leq \mathbf{tw}(G) + 1$. Comme chaque composante connexe de G' est 2-connexe, par la Proposition 3 il nous suffit de prouver que $\mathbf{tw}(C') \leq \mathbf{tw}(C) + 1$, où C' une composante connexe de G et C sa composante associée dans G .

La décomposition arborescente de G est l'union des décompositions arborescentes de chacune de ses composantes connexes. Soient une composante connexe C de G et une décomposition arborescente optimale $(T^{\text{opt}}, \mathcal{X}^{\text{opt}})$ de C . Si x est un sommet d'articulation de C , nous nommerons $\{C_1, \dots, C_{p(x)}\}$ les composantes connexes de $C - x$.

Alors, si pour $i \in [p(x)]$, nous notons $(T_i^{\text{opt}}, \mathcal{X}_i^{\text{opt}})$ la restriction de $(T^{\text{opt}}, \mathcal{X}^{\text{opt}})$ au sous-graphe C_i de C , nous pourrions toujours trouver une décomposition arborescente optimale (T, \mathcal{X}) de C telle que l'intersection de deux sacs quelconque entre \mathcal{X}_i et \mathcal{X}_j , pour $i \neq j \in [p(x)]$, soit réduite à $\{x\}$, ou vide. Si C' est la composante connexe de G' obtenue en appliquant l'Algorithme 1 sur C , nous pouvons construire une décomposition arborescente (T', \mathcal{X}') de C' à partir de (T, \mathcal{X}) comme suit.

Pour un sommet d'articulation x de C donné, nous noterons $\{C'_1, \dots, C'_{p(x)}\}$ les composantes connexes de $C' - x$. Nous allons alors construire les décompositions arborescentes (T'_i, \mathcal{X}'_i) de C_i et considérerons (T'_i, \mathcal{X}'_i) comme la restriction de (T', \mathcal{X}') à C_i , pour chaque $i \in [p(x)]$. Soit l'ensemble $\{x_1, \dots, x_{p(x)}\}$, composé des sommets du cycle ou du chemin remplaçant x dans C' et fixons un $i \in [p(x)]$. Soient alors a_{i_1} et a_{i_2} comme dans l'algorithme, nous posons :

- $V(T'_i) = V(T_i)$, $E(T'_i) = E(T_i)$ et $\mathcal{X}'_i = \mathcal{X}_i$,
- $\forall X \in \mathcal{X}'_i$:
 - si $x \in X$, alors $X = (X \cup \{x_i\}) \setminus \{x\}$,
 - sinon si x et $a_{i_2} \in X$, alors $X = (X \cup \{x_{i+1}\}) \setminus \{x\}$.

Soit ensuite (T_x, \mathcal{X}_x) , une décomposition arborescente optimale de l'ensemble $\{x_1, \dots, x_{p(x)}\}$. Sa largeur est d'au plus deux. Alors, pour construire (T', \mathcal{X}') , nous faisons :

- $\mathcal{X}' = \mathcal{X}_x \cup \bigcup_{i=1}^{p(x)} \mathcal{X}'_i$ et $V(T') = V(T_x) \cup \bigcup_{i=1}^{p(x)} V(T'_i)$,
- $E(T') = E(T_x) \cup \{\{X, X_{x_i}\} \mid X \in \mathcal{X}', X_{x_i} \in \mathcal{X}_x \text{ et } x_i \in X \cap X_{x_i}\} \cup \bigcup_{i=1}^{p(x)} E(T'_i)$.

La taille de chaque sac $X \in \mathcal{X}'$ est la même que celle du sac qui lui est associé dans \mathcal{X} . Et celle de chaque sac dans \mathcal{X}_x est d'au plus deux, car $\{x_1, \dots, x_{p(x)}\}$ est un cycle ou un chemin. Il nous reste à prouver que (T', \mathcal{X}') est bien une décomposition arborescente de C' .

En effet, une arête de C' est de trois formes, soit elle est interne à un C'_i , pour un sommet d'articulation x de C donné, soit elle est interne au sous-graphe induit par l'ensemble $\{x_1, \dots, x_{p(x)}\}$, soit enfin elle possède une extrémité dans un C'_i et une autre

dans $\{x_1, \dots, x_{p(x)}\}$. Le premier cas est couvert par le fait que (T, \mathcal{X}) était bien une décomposition arborescente de C et le second par construction de (T_x, \mathcal{X}_x) . Enfin une arête $\{s, x_j\}$ de C' dans le troisième cas implique, par construction de l'algorithme, qu'il existait une arête $\{s, x\}$ dans C . Ce qui est couvert par le deuxième point de la construction de la décomposition arborescente (T'_i, \mathcal{X}'_i) de C'_i .

Enfin, pour tout $s \in V(C)$, il nous reste à vérifier que l'ensemble des $t \in V(T')$ tels que $s \in X_t \in \mathcal{X}'$ forme un sous-arbre de T' . Deux cas s'offrent à nous. Lorsque $s \in C'_i$, pour un $i \in [p(x)]$ et pour un sommet d'articulation x de C donnés, aucun des sacs de $\mathcal{X}' \setminus \mathcal{X}'_i$ ne contient s . Et, lorsque $s \in \{x_1, \dots, x_{p(x)}\}$, les sacs contenant s sont les sacs de \mathcal{X}'_i tels que leurs sacs associés dans \mathcal{X}_i contenaient x . La preuve du second cas découle du fait que l'ensemble des sommets de T , dont les sacs correspondant dans \mathcal{X} contiennent x , forment un sous-arbre de T .

Ainsi, la largeur de la décomposition arborescente (T', \mathcal{X}') de C' est soit égale à celle de (T, \mathcal{X}) , soit, si C est un arbre, égale à deux. ■

2.2.3 L'augmentation radiale

Soit maintenant G un graphe lisse. Rappelons que pour qu'une grille annulaire simule l'ajout de k arêtes potentielles dans n'importe qu'elle face de G , il est nécessaire que la frontière de cette face soit un cycle. Mais comme nous supposons plus généralement que G n'est pas un graphe connexe, par la Proposition 8, il nous suffit de connecter entre elles les composantes 2-connexes de G afin que le graphe résultant soit 2-connexe.

Choix d'une carte planeaire

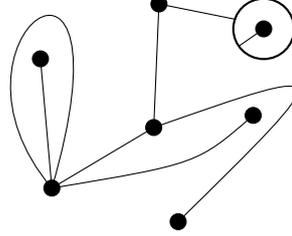
Connecter les composantes 2-connexes de G entre elles, c'est les connecter entre elles face de G par face de G . Soient F_1, \dots, F_f les faces de G et $i \in [f]$. Comme G est lisse, la frontière de F_i est une union de cycles. Considérons alors le graphe dont les sommets sont les cycles de la frontière de F_i . Notre but est de connecter entre eux les sommets de ce graphe.

Pour cela, nous allons utiliser la notion de carte planeaire. Une carte planeaire étant déjà connexe, cela garantira la connexité. L'idée derrière la nécessité de l'utilisation des cartes planeaires, et non d'une structure plus simple comme les arbres couvrants, est dû au fait qu'un choix d'arête connectant deux cycles de la frontière de F_i peut bloquer, à cause de la contrainte de planarité, un ajout ultérieur d'arête permettant de réduire le diamètre du graphe. Nous devons donc tester tous les choix de cartes planeaires possibles. Notons que les cartes planeaires doivent avoir au plus k arêtes puisque nous ne pouvons ajouter que k arêtes par face.

Ainsi, posons \mathcal{S}_i comme l'ensemble de toutes les cartes planeaires dont les sommets sont les cycles de F_i , voir Figure 2.4. Et soit

$$\mathbf{S}_G^{(k)} = \{(S_1, \dots, S_f) \in \mathcal{S}_1 \times \dots \times \mathcal{S}_f \mid \forall i \in \{1, \dots, f\} |E(S_i)| \leq k\},$$

l'ensemble des choix possibles d'une carte planeaire d'au plus k arêtes dans chacune des faces de G . Chaque choix induit par un élément de $\mathbf{S}_G^{(k)}$ rend donc G connexe. Le lemme suivant nous donne le nombre de choix de cartes planeaires possibles en fonction de G .

FIGURE 2.4 – Carte planaire simple sur les cycles d’une face de G .

Le théorème suivant va nous permettre de compter tous les choix possibles de cartes planaires à l’intérieur d’une même face de G .

Théorème 11 ([Albenque and Poulalhon, 2013])

Le nombre de cartes planaires à k arêtes est de

$$\frac{2 \cdot 3^k}{(k+2)(k+1)} \binom{2k}{k} \sim 4^k.$$

Enfin, en corollaire, nous pouvons donner la taille de l’ensemble $\mathbf{S}_G^{(k)}$.

Corollaire 3

Soit G un graphe lisse, alors $|\mathbf{S}_G^{(k)}| \leq (4^k)^n$.

Démonstration : Par la formule d’Euler, nous savons que le nombre de faces d’un graphe plan est d’au plus n . Et le nombre de choix possibles de cartes planaires dans une face est d’au plus 4^k . De plus, $\mathbf{S}_G^{(k)}$ est l’ensemble des vecteurs de choix possibles d’une carte planaire d’au plus k arêtes dans chacune des faces de G . Chaque coordonnée d’un vecteur correspondant à une face différente de G . Le nombre de ces vecteurs est donc multiplicatif, i.e. $|\mathbf{S}_G^{(k)}| \leq (4^k)^n$. ■

Comme il semble difficile, voir impossible de borner le nombre de faces par une fonction de d et de k , toute velléité d’algorithme FPT paramétré par d et k s’effondre. Cependant, si nous nous restreignons au problème BCP, i.e. tel que le nombre d’arêtes ajoutées dans tout le graphe est d’au plus k , alors par amortissement, le nombre de cartes planaires possibles dans tout le graphe G sera d’au plus 4^k . Dans ce cas, nous définissons $\mathbf{S}_G^{(k)}$ alternativement comme suit.

$$\mathbf{S}_G^{(k)} = \{(S_1, \dots, S_f) \in \mathcal{S}_1 \times \dots \times \mathcal{S}_f \mid \sum_{i \in \{1, \dots, f\}} |E(S_i)| \leq k\},$$

Et ainsi, nous avons un corollaire alternatif au Théorème 11.

Corollaire 4

Lorsque le nombre total d'arêtes ajoutée est d'au plus k , alors $|\mathbf{S}_G^{(k)}| \leq 4^k$.

Nous allons quand même présenter une programmation dynamique pour le problème BFCP, car il est plus général et que nous avons de l'espoir de trouver une autre méthode permettant de connecter le graphe sans recourir à un outil aussi volumineux que l'ensemble des cartes planaires possibles.

Proposition 11

Soit G un graphe plan. Si le nombre de composantes connexes $\Delta(G)$ de G est strictement plus grand que $k + 1$, alors $\text{BCP}(G) = \text{Faux}$.

Démonstration : Supposons que $\Delta(G) > k+1$. Alors pour que le diamètre de la complétion plane G^+ de G soit borné, il faut que G^+ soit connexe. Mais pour rendre G^+ connexe, il nous faudrait connecter entre elles toutes les composantes connexes de G . Considérons alors le graphe dont les sommets sont les composantes connexes de G . Ce graphe est donc réduit à un ensemble de $\Delta(G)$ sommets indépendants. Pour connecter ce graphe, le nombre minimum d'arêtes requises est le nombre d'arêtes d'un arbre couvrant d'un graphe à $\Delta(G)$ sommets, i.e. $\Delta(G) - 1 > k$ arêtes. Ce qui implique que $\text{BCP}(G) = \text{Faux}$. ■

L'augmentation

Afin de plonger dans chaque face de G la carte planaire choisie qui lui correspond, nous allons ajouter une grille annulaire particulière autour de chaque cycle de la frontière de la face puis nous connecterons les grilles annulaires entre elles en dupliquant chaque pont ainsi créer afin de rendre le graphe final 2-connexe. Tout d'abord présentons $\Gamma_{k,r,s}$, la grille annulaire transformée, permettant d'assurer les connexions entre deux grilles annulaires.

Soit s un entier tel que $r \geq s \geq 3$ et considérons un ensemble Q de $r - s$ sommets consécutifs dans la base de $\Gamma_{k,r}$. Soit alors $\Gamma_{k,r,s}$, le graphe obtenu en supprimant toutes les arêtes ne se trouvant pas dans la base de $\Gamma_{k,r}$ et ayant au moins une extrémité dans Q , puis en remplaçant le chemin de la base, formé par les sommets de Q , par une unique arête entre les deux extrémités du chemin.

De plus, le toit de $\Gamma_{k,r,s}$ est le même que le toit de $\Gamma_{k,r}$ et sa base est la nouvelle frontière de f_1 dans $\Gamma_{k,r,s}$. Nous présentons maintenant, d'une manière moins informelle, comment les grilles annulaires transformées sont placées autour des cycles composant la frontière de chaque face de G . Pour une face F_i de G , nous dénoterons par ρ_i le nombre de cycles composant sa frontière.

Soit $\bar{S} = (S_1, \dots, S_f) \in \mathbf{S}_G^{(k)}$ un choix possible d'une carte planaire dans chaque face de G . Nous allons construire le graphe $G_{\bar{S}}$ comme suit. Soient F_i une face de G et $C_1^i, \dots, C_{\rho_i}^i$ les cycles composant sa frontière. Pour tout cycle C_j^i , avec $j \in [\rho]$, considérons une courbe fermée Λ_j^i dans l'intérieur de F telle que le disque ouvert borné par Λ_j^i , parmi les deux disques ouverts définis par Λ_j^i , contienne C_j^i et que le deuxième contienne les autres cycles de la frontière de F , s'ils existent.

Nous ajoutons alors une copie Γ_j^i de $\Gamma_{k, \max\{|C_j^i|, k\}, |C_j^i|}$ dans l'intérieur de F_i tel que le toit de Γ_j^i soit identifié à Λ_j^i et sa base à C_j^i . Remarquons qu'il y a $|C_j^i|$ manière d'identifier la base de $\Gamma_{k, \max\{|C_j^i|, k\}, |C_j^i|}$ avec C_j^i , selon le choix de l'ensemble Q , et nous en choisissons alors une arbitrairement. Nous appliquons alors cette construction pour chaque face F_i de G et dénotons $G_{\bar{S}}$ le graphe lisse obtenu. Nous devons ensuite connecter les grilles transformées entre elles. Cela se fait de la manière suivante.

Définissons pour un $i \in [f]$, le graphe \hat{S}_i comme suit. Considérons le plongement dans le plan du graphe L' induit par un choix d'une carte planaire de S_i et dupliquons tous ses ponts. Dans le plongement résultant L , nous prenons une collection de courbes fermées $\{K_j^i \mid j \in [\rho_i]\}$ telle que chaque K_j^i définisse un disque fermé Δ_j^i dans lequel $L \cap \Delta_j^i$ soit le plongement dans le plan d'un graphe étoilé avec le sommet C_j^i pour centre.

Alors nous posons l'ensemble des points du plans définis par $D_j^i = \Delta_j^i \cap S_i$ comme les sommets de \hat{S}_i et les composantes connexes de $\Delta_j^i \setminus D_j^i$ comme les arêtes de \hat{S}_i . Dans le plongement résultant, nous retirons ensuite tous les sommets dans $V(S_i)$ ainsi que toutes leurs arêtes adjacentes. Remarquons que chaque K_j^i forme maintenant un cycle. La construction de \hat{S}_i se termine en choisissant arbitrairement une arête dans chaque K_j^i et en la remplaçant par un chemin de manière à ce que le cycle résultant, noté \hat{K}_j^i soit de taille $\max\{|C_j^i|, k\}$.

Enfin, nous définissons le graphe $\tilde{G}_{\bar{S}}$ comme suit. Considérons le graphe plan résultant de la fusion des graphes $G_{\bar{S}}$ et $\hat{S}_i, i \in [f]$ telle que, pour chaque $j \in [\rho_i]$, le toit de Γ_j^i soit identifié avec le cycle \hat{K}_j^i de \hat{S}_i .

S'en suit la définition d'une augmentation radiale.

Définition 9 (*Augmentation radiale*)

Soit G un graphe lisse. Alors le graphe $\tilde{G}_{\bar{S}}$, ainsi obtenu à partir de G et pour un choix de cartes planaires $\bar{S} \in \mathbf{S}_G^{(k)}$, s'appelle une augmentation radiale de G .

Conséquences

À l'instar du Lemme 1, où nous avons prouvé que la largeur arborescente d'un graphe lissé restait linéaire en celle du graphe de départ. Le lemme suivant énonce que la largeur arborescente d'une augmentation radiale de G n'augmente que d'un facteur k . Ce résultat ne s'applique que si nous considérons le problème B lorsque le nombre total d'arêtes ajoutées est borné par k .

Lemme 2

Soit G un graphe lisse et soit $H = \tilde{G}_{\bar{S}}$ pour un $\bar{S} \in \mathbf{S}_G^{(k)}$, alors

1. $|V(H)| = O(k \cdot n)$,
2. $\mathbf{tw}(H) = O(k \cdot \mathbf{tw}(G))$.

Démonstration : 1. Une grille annulaire transformée, du type $\Gamma_{k,r,s}$, est ajoutée autour de chaque cycle de la frontière de toutes les faces de G . Alors, pour un tel cycle fixé de taille r , le nombre de sommet de son augmentation est de $k \cdot r$. Posons f comme le nombre de faces de G et ρ_i comme le nombre de cycles de la frontière de la $i^{\text{ème}}$ face de G . Puisque tout sommet de H appartient à un tel cycle, car H est 2-connexe, nous pouvons alors écrire

$$|V(H)| = \sum_{i \in [f]} \sum_{j \in [\rho_j]} k \cdot r_j = k \cdot \sum_{i \in [f]} \sum_{j \in [\rho_j]} r_j = k \cdot |V(G)|.$$

2. Fixons un $i \in [f]$ et un $j \in [\rho_i]$ et soit C_j^i un cycle de la frontière de la face F_i de G . Soit alors R_U le graphe radial du sous-graphe U de G induit par le cycle C_j^i . Par la Proposition 4, nous savons que $\mathbf{tw}(U + R_U) \leq 2 \cdot \mathbf{tw}(U)$. Soit ensuite M_{U+R_U} , le graphe médial du graphe $U + R_U$. Par le Corollaire 1, nous avons $\mathbf{tw}(U + R_U + M_{U+R_U}) \leq 2 \cdot \mathbf{tw}(U + R_U)$, donc $\mathbf{tw}(U + R_U + M_{U+R_U}) \leq 4 \cdot \mathbf{tw}(U)$. Remarquons ensuite que le cycle C_j^i et la première couche de sa grille annulaire est un sous-graphe de $U + R_U + M_{U+R_U}$ donc sa largeur arborescente est d'au plus quatre fois celle de U .

Soit alors une décomposition arborescente optimale (T, \mathcal{X}) du cycle C_j^i avec la première couche de sa grille annulaire. Nous allons maintenant ranger tous les autres sommets de la grille annulaire dans les sacs de \mathcal{X} . Pour ce faire, nous procédons comme suit. Pour chaque sac $X \in \mathcal{X}$ et pour tout $x \in X$, nous ajoutons dans X la ligne entière de x . Les nouveaux sacs X' ainsi obtenus définissent l'ensemble \mathcal{X}' . Et si nous posons $T' = T$, nous avons une décomposition arborescente de C_j^i augmenté, i.e. avec sa grille annulaire.

En effet, soit une arête $\{x, y\}$ de C_j^i augmenté. Si c'est une arête de C_j^i ou de la première couche de sa grille annulaire, alors il existe déjà un sac $X \in \mathcal{X}$ tel que $\{x, y\} \subseteq X$. Sinon, deux cas s'offrent à nous. En premier lieu, si les deux sommets sont dans une même couche de la grille annulaire, alors x_1 , le sommet de la ligne L_x dans la première couche, et y_1 , le sommet de la ligne L_y dans la première couche, sont adjacents. Et comme (T, \mathcal{X}) est une décomposition arborescente du cycle C_j^i avec la première couche de sa grille annulaire, il existe un sac $X \in \mathcal{X}$ contenant x_1 et y_1 . Alors par construction, L_x et $L_y \subseteq X'$, donc x et $y \in X'$.

Ensuite, soit x un sommet de C_j^i augmenté et soient $T_x = \{X' \in \mathcal{X}' \mid x \in X'\}$. Alors T_x est un sous-arbre de T' . En effet, comme $T' = T$, nous avons $T_x = \{X \in \mathcal{X} \mid x_1 \in X\}$, si x_1 est le sommet de la première couche de L_x , qui par définition est un sous-arbre de T .

Nous avons donné les décompositions arborescentes de toutes les grilles annulaires de H . Nous allons construire une décomposition arborescente de H à partir des décompositions de chaque cycle augmenté.

Nous devons maintenant inclure les arêtes ajoutées via les choix de cartes planaires. Mais dans le cas où nous ne nous intéressons qu'au problème où le nombre total d'arêtes ajoutées est d'au plus k , les arêtes ajoutées via les choix de cartes planaires sont d'au plus k dans l'ensemble du graphe. Et par la Proposition 2 nous savons qu'à chaque fois qu'une arête est ajoutée dans un graphe, la largeur arborescente n'augmente qu'au plus de un. Donc si dans le pire des cas, toutes les arêtes ajoutées par le choix des cartes planaires sont doublées, afin de conserver la biconnexité, la largeur arborescente n'augmentera que de $2 \cdot k$, ce qui reste linéaire en k .

Ainsi, la largeur arborescente de H est en $O(k \cdot \text{tw}(G))$. ■

2.3 La (d, k, s, w) -arête colorabilité

Dans la suite, sauf mention du contraire et jusqu'à ce que nous discutons une nouvelle fois de complexité, nous considérerons le problème plus général BFCP, où le nombre d'arêtes par face est d'au plus k .

2.3.1 Définition

Définition 10 ((d, k, s, w) -arête colorabilité)

Soient G et H deux graphes plans tels que G soit un sous-graphe de H .

Posons $E^{\text{old}} = E(G)$ et $E^{\text{new}} = E(H) \setminus E(G)$.

Et soient $k, d, s \in \mathbb{N}$ et $w : E^{\text{old}} \rightarrow \mathbb{N}$ une fonction de poids sur E^{old} .

E^{new} est alors dit (d, k, s, w) -arête colorable dans H , lorsqu'il existe une 3-partition $\mathbf{p} = \{E^0, E^s, E^\infty\}$ de E^{new} telle que si $w_{\mathbf{p}} = w \cup \{(e, 0) \mid E \in E^0\} \cup \{(e, s) \mid E \in E^s\} \cup \{(e, d+1) \mid E \in E^\infty\}$, alors les trois propriétés suivantes sont vérifiées :

- (1) toute face F de G contient au plus k arêtes de E^s ,
- (2) $\forall x, y \in V^{\text{old}}, \exists (x, y)$ -chemin P dans H tel que $w_{\mathbf{p}}(E(P) \cap (E^{\text{old}} \cup E^s)) \leq d$,
- (3) $\forall x, y \in V^{\text{old}} \neg \exists (x, y)$ -chemin P contenant uniquement des arêtes de E^0 .

Remarquons que le fait que nous ayons besoin d'une fonction de poids sur les arêtes du graphe de départ G est dû au fait que nous avons besoin que G soit lisse et donc il peut exister un ensemble d'arêtes $J \subseteq E(G)$.

Notons que nous pouvons définir une notion plus simple de la (d, k, s, w) -arête colorabilité pour le problème BCP, i.e. lorsque le nombre total d'arêtes ajoutées est d'au plus k . En effet, il nous suffit de remplacer la propriété (1) par la propriété suivante :

(1.bis) H contient au plus k arêtes de E^s .

2.3.2 Équivalence entre BFCP et la (d, k, s, w) -arête colorabilité

Couplages non-croisés d'un cycle et dual faible

Soit alors C un graphe plan réduit à un cycle et posons $|C| = h \in \mathbb{N}$. La définition suivante va formaliser la notion d'ajout d'arêtes dans une face.

Définition 11 (*Couplage partiel non-croisé d'un cycle*)

Un couplage partiel non-croisé M de C est un ensemble de paires de $V(C)$ telles que chaque paire est composée de deux sommets non-adjacents dans $V(C)$ et telles que si nous dessinons dans l'intérieur de C les arêtes induites par ces paires, aucun croisement ne serait créé.

Soit M un couplage partiel non-croisé de C . Posons alors $C + M = (V(C), E(C) \cup M)$. Puisque les paires de M induisent des arêtes non-croisées, $C + M$ est un graphe plan extérieur biconnexe.

Les faces simpliciales nous seront d'une grande aide pour formaliser la simulation de l'ajout d'arêtes dans une face par la grille annulaire.

Définition 12 (*Face simpliciale*)

Soit M un couplage partiel non-croisé de C . Une face de $C + M$ est dit simpliciale lorsque son intersection avec C forme une composante connexe de C .

Notre preuve se fera par induction. La structure d'arbre du dual faible nous sera alors précieuse.

Définition 13 (*Dual faible*)

Le dual faible d'un graphe plan G est le graphe dont les sommets correspondent aux faces bornées de G et tel que deux sommets sont adjacents si les frontières de leurs faces respectives dans G admettent une arête de G en commun. Le dual faible de G est noté par G_w^* , voir la Figure 2.5.

La notion de profondeur nous permettra d'asseoir l'induction sur les nombres entiers.

Définition 14 (*Profondeur d'un arbre*)

La profondeur d'un arbre est définie de manière inductive.

- l'arbre réduit à un sommet unique admet une profondeur nulle.
- Soient T et T' deux arbres tels qu'en supprimant toutes les feuilles de T' nous obtenions T , alors la profondeur de T est égale à la profondeur de T' moins un.

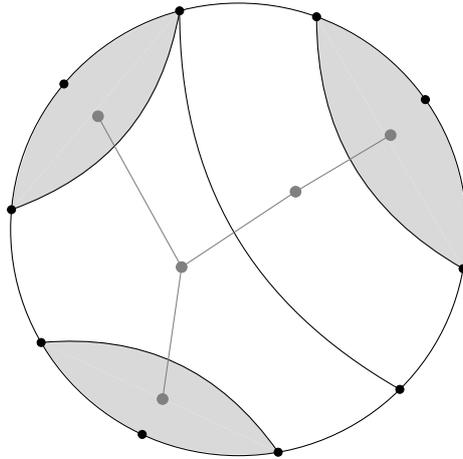


FIGURE 2.5 – Un graphe planaire extérieur 2-connexe G , son dual faible G_w^* et ses faces simpliciales.

Une propriété intéressante concernant notre problème est la suivante.

Proposition 12

Le dual faible d'un graphe plan extérieur et 2-connexe G est toujours un arbre.

Démonstration : Un cycle dans T impliquerait un sommet intérieur dans G . Comme G est 2-connexe, ce sommet intérieur et un sommet quelconque de la frontière de la face extérieure appartiendraient à un cycle C' de G . Mais ce cycle serait alors la frontière d'une face, dont deux sommets au moins sont adjacents et qui n'est pas externe. Ce qui contredirait le fait que G est un graphe plan extérieur. ■

Simulation

Nous allons maintenant montrer comment une grille annulaire augmentant un cycle peut simuler l'ajout d'arêtes à l'intérieur de ce cycle. Dans la suite, M sera un couplage partiel non-croisé de C de taille k . Comme le dual faible de $C + M$ est un arbre, il est composé soit d'un unique sommet, soit possède au moins deux feuilles. Ces feuilles là correspondent alors aux faces simpliciales de C . Soit $\Gamma_{k,r}$ une grille annulaire de C , nous noterons $C + \Gamma$ l'ajout de $\Gamma_{k,r}$ dans C comme décrit dans la sous-section précédente discutant des grilles annulaires.

Le lemme suivant est au cœur de la simulation.

Lemme 3

$C + M$ est isomorphe à un mineur de $C + \Gamma$.

Remarquons qu'il est suffisant de trouver une 3-partition $\{E_\ell^0, E_\ell^1, E_\ell^\infty\}$ des arêtes de chaque ℓ -couche, $\ell \in [k]$, de $C + \Gamma$ telle qu'en contractant les arêtes de E^0 et en supprimant les arêtes de E^∞ dans chaque couche $\ell \in [k]$, nous obtenions le graphe $C + M$, i.e. $C + M$ sera un mineur de $C + \Gamma$. Nous voulons de plus que dans cette simulation, les trois points de la (d, k, s, w) -arête colorabilité soient garantis.

La preuve de ce lemme est basée sur une induction sur la profondeur du dual faible de $(C + M)_w^*$ de $C + M$. Au départ, nous déclarons l'arbre $T_1 = C + M_w^*$, et à chaque étape de l'induction, nous considérons les feuilles de T_1 puis nous les supprimons. La profondeur de l'arbre résultant T_2 va décroître de un, nous continuons donc l'induction jusqu'à ce que la profondeur soit nulle. Chaque feuille correspond à une face simpliciale de $C + M$ et en particulier à l'arête interne bornant cette face. Supprimer les arêtes de T_ℓ à la fin de l'étape ℓ correspondra alors à la suppression de ces arêtes internes bornant les faces simpliciales associées aux feuilles. Ainsi, pour chaque étape ℓ , nous définissons le graphe plan extérieur 2-connexe C_ℓ dont le dual faible est l'arbre T_ℓ . En particulier, $C_1 = C + M$.

La notion de degré interne ne sert qu'à nous simplifier la tâche.

Définition 15 (*ℓ -degré interne*)

Soit x un sommet de C . Le ℓ -degré interne de x , noté $\mathbf{di}_\ell(x)$, représente le nombre d'arêtes internes de C_ℓ adjacentes à x .

Voici la preuve du Lemme 3.

Démonstration: Soient $\{x_1, \dots, x_h\}$ les sommets de C et $\{x_\ell^1, \dots, x_\ell^h\}$ les sommets de la ℓ -couche, pour $\ell \in [k]$. Et supposons que l'induction en soit à l'étape $\ell \in [k]$. Soit alors a une feuille de T_ℓ associée à l'arête interne $e_a = \{x_s, x_t\}$ de C_ℓ . La feuille a va correspondre au plus court chemin dans la ℓ -couche entre x_s et x_t . Ce plus court chemin nous donnera une 3-partition des arêtes de la ℓ -couche via l'Algorithme 3.

À chaque étape ℓ de l'induction, supposons que nous avons traité les arêtes internes e_1, \dots, e_p de $C + M$. Alors en supprimant les arêtes de E_ℓ^∞ et en contractant celles de E_ℓ^0 , nous obtenons un mineur \mathfrak{M} de $C + \Gamma$. De plus, l'Algorithme 3 nous garantit que :

1. pour tout $\{x, y\} \in \{e_1, \dots, e_p\}$ une arête entre x et y est créée dans \mathfrak{M} ,
2. aucun croisement d'arêtes n'est créé,
3. aucun sommet interne n'est créé,

1. L'Algorithme 3 nous garantit un chemin entre x et y composé uniquement d'arêtes de E_ℓ^0 et d'une arête de E_ℓ^s .

2. Étant donné deux paires $\{x, y\}$ et $\{u, v\}$ de sommets extrémités de deux arêtes internes de $C + M$ telles que le plus court chemin entre x et y dans la ℓ -couche traverse celui entre u et v . Cependant, les seules possibilités pour qu'une telle configuration puisse exister sont lorsque $|C| = 4$ ou lorsque au moins une des deux faces de $C + M$ induites par $\{x, y\}$ ou $\{u, v\}$ n'est pas simpliciale. Le premier cas est impossible puisqu'alors une seule arête interne peut être dessinée dans l'intérieur de C . Et le second cas est une contradiction avec le fait que $\{x, y\}$ et $\{u, v\}$ sont les extrémités de deux arêtes internes.

3. L'existence d'un sommet interne implique l'existence d'au moins deux arêtes adjacentes dans E_ℓ^s . Mais l'algorithme empêche une telle configuration, puisque nous choisissons les arêtes de E_ℓ^s de sorte à ce qu'elles se trouvent toujours seule dans un chemin d'arêtes contractées entre deux sommets de C . ■

Entrées : C_ℓ, e_1, \dots, e_p les arêtes internes de C_ℓ et la ℓ -couche.

Output : $(E_d^\ell, E_c^\ell, E_r^\ell)$ une 3-partition des arêtes de la ℓ -couche.

début

```

    SE := {e1, ..., ep}
    tant que SE ≠ ∅ faire
        Choisir {xs, xt} ∈ SE
        Trouver le plus court chemin P dans la ℓ-couche entre xs et xt
        si |E(P)| > 2 alors
            pour i ∈ [h] faire
                si xiℓ ∉ ej ∀ j ∈ [p] alors
                    | {xiℓ-1, xiℓ} ∈ Edℓ
                sinon
                    | {xiℓ-1, xiℓ} ∈ Ecℓ
                fin
            fin
            Choisir e ∈ E(P) \ (Edℓ ∪ Ecℓ) and put e ∈ Erℓ
        sinon
            {xsℓ-1, xsℓ} ∈ Erℓ
            {xtℓ-1, xtℓ} ∈ Ecℓ
        fin
        si |E| = 1 alors
            pour e ∈ E(ℓ-couche) \ (Ecℓ ∪ Erℓ ∪ Edℓ) faire
                | e ∈ Edℓ
            fin
        fin
        SE := SE \ {e}
    fin
    retourner (Edℓ, Ecℓ, Erℓ)
fin

```

Algorithme 3: partitionnement

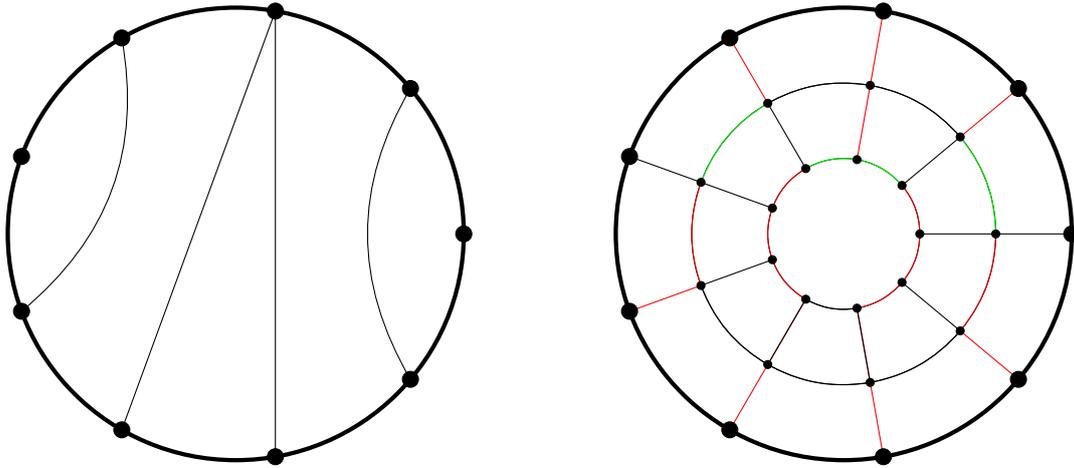


FIGURE 2.6 – $C + M$ et une 3-partition des arêtes de $C + \Gamma$. Les arêtes rouges sont les arêtes de E^0 , les noires celles de E^s et les vertes celles de E^∞ . Une arête interne de $C + M$ est associée à chemin rouge-vert dans $C + \Gamma$.

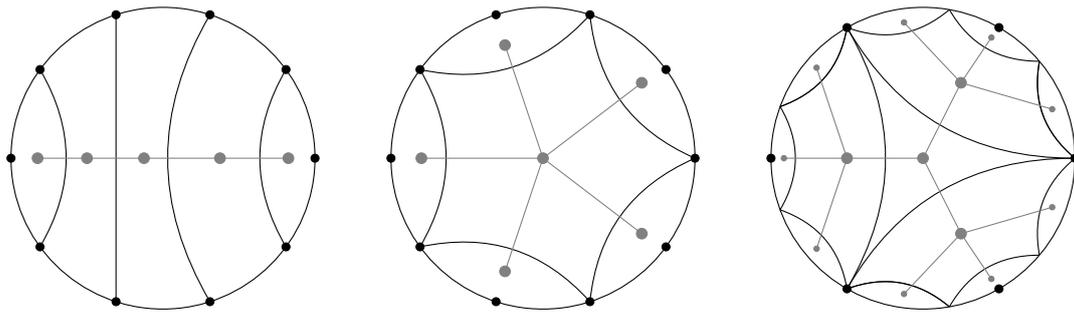


FIGURE 2.7 – $C + M$ dont le dual faible est : un chemin (gauche), une étoile (milieu) un arbre ternaire (droite).

La profondeur d'un arbre de k sommets est calculable par induction en un nombre d'étapes linéaire en k . Le pire des cas étant P_k , le chemin de longueur k , qui a une profondeur de $\frac{k}{2}$, alors que le meilleur des cas est le graphe étoilé qui a une profondeur de un. Notons que lorsque un arbre est d'arité ℓ , sa profondeur est de $\log_\ell(k)$. Nous pouvons relier ces observations sur la profondeur de $T\ell$ avec les configurations possibles d'arêtes internes de C_ℓ . Ainsi, la Figure 2.7 illustre ce lien.

Remarquons que nous pourrions nous contenter d'une grille annulaire de $\frac{k}{2}$ couches pour simuler l'ajout de k arêtes dans n'importe quelle face, mais que nous en utilisons une avec k couches. Cela n'est pas optimal mais permet de ne pas se soucier de détails particuliers comme la parité de k .

L'équivalence

Soit le problème suivant.

ARÊTE SOMMET-COMPLÉTION PLANE À DIAMÈTRE BORNÉ (ASCPDB)
 Entrée : un graphe plan G , $d, k \in \mathbb{N}$.
 Question : existe-t-il au plus k arêtes de $\binom{V}{2}$ et des nouveaux sommets, définis comme des points de $\mathbb{R}^2 \setminus V(G)$, à ajouter par face de G tels que le diamètre du graphe plan résultant soit d'au plus d ?

Alors nous avons la proposition suivante.

Proposition 13

Soient G un graphe lisse et $d, k \in \mathbb{N}$, alors

$$ASCPDB(G, d, k) = \text{Vrai} \iff BF(G, d, k) = \text{Vrai}.$$

Démonstration : Comme le problème ASCPDB est plus général que le problème BFCP, puisqu'il englobe le cas où aucun nouveau sommet n'est ajouté dans aucune face de G , si (G, d, k) est une instance positive de BFCP, alors c'est aussi une instance positive de ASCPDB.

Réciproquement, supposons que (G, d, k) soit une instance positive de ASCPDB. Alors, en contractant les arêtes de la complétion G' de G du problème ASCPDB ayant un nouveau sommet à chaque extrémité, nous obtenons une instance complétée G'' de G du problème BF. Comme G' est une complétion positive, elle possède au plus k arêtes ajoutées par face et admet un diamètre d'au plus d . Les arêtes contractées permettant de passer de G' à G'' sont toutes des arêtes ajoutées puisqu'elles sont incidentes à des nouveaux sommets. Donc en contractant ces arêtes, nous avons réduit le nombre d'arêtes ajoutées par face et leur nombre reste donc bien inférieur ou égal à k . De même, en contractant des arêtes d'un graphe, nous réduisons son diamètre. Ainsi (G, d, k) est bien une instance positive de BFCP. ■

Le lemme suivant nous donne l'équivalence entre notre problème et la (d, k, s, w) -arête colorabilité.

Lemme 4

Soient G_{input} un graphe plan et G son graphe lissé comme dans l'Algorithme 1, avec J l'ensemble des arêtes des cycles de G remplaçant les sommets d'articulation de G_{input} . Soient ensuite $d, k \in \mathbb{N}$ et $J \subseteq E(G)$. Alors il existe une complétion plane G^+ de G_{input} , dont le diamètre est d'au plus d , si et seulement si pour un $\bar{s} \in \mathbf{S}_G^{(k)}$, l'ensemble d'arêtes $E(\tilde{G}_{\bar{s}}) \setminus E(G)$ est (d, k, s, w_J) -arête colorable dans $\tilde{G}_{\bar{s}}$, où $s = 1$ et $w_J = \{(e, \langle e \notin J \rangle) \mid e \in E(G)\}$.

Nous posons alors les notations suivantes :

- $E^{\text{old}} = E(G)$, $V^{\text{old}} = V(G)$,
- $E^{\text{new}} = E(H) \setminus E(G)$, $V^{\text{new}} = V(H) \setminus V(G)$.

Voici la démonstration du Lemme 4.

Démonstration : 1. Supposons qu'il existe une augmentation radiale $\tilde{G}_{\bar{S}}$ de G pour un $\bar{S} \in \mathbf{S}_G^{(k)}$ et que $E(\tilde{G}_{\bar{S}}) \setminus E(G)$ soit (d, k, s, w_J) -arête colorable. Puisque $E(\tilde{G}_{\bar{S}}) \setminus E(G)$ est (d, k, s, w_J) -arête colorable, il existe une 3-partition $\mathbf{p} = \{E^0, E^s, E^\infty\}$ de $E(\tilde{G}_{\bar{S}}) \setminus E(G)$ vérifiant les axiomes de la Définition 2.3. Alors, en supprimant les arêtes de E^∞ et en contractant les arêtes de E^0 et de J , nous obtenons un graphe plan G' . Alors il existe une complétion plane de G à diamètre borné et ne possédant qu'au plus k arêtes ajoutées par face de G .

En effet, par la Proposition 13, nous pouvons contracter toutes les arêtes de G' dont les extrémités sont des sommets de E^{new} . Le graphe résultant sera noté G^+ . Soit alors une paire de sommets x, y de $V(G) = V(G^+)$. Ce sont par construction des sommets de V^{old} . Par le point (3) de la (d, k, s, w_J) -arête colorabilité de $E(\tilde{G}_{\bar{S}}) \setminus E(G)$, il existe un chemin P dans $\tilde{G}_{\bar{S}}$ entre les sommets x et y et par le point (2), le poids de P est d'au plus d . Puisque ce chemin P est de poids au plus d , il n'emprunte aucune arête de E^∞ , qui elles ont un poids de $d + 1$. De plus, comme les arêtes de E^0 et de J ont un poids égal à zéro, les seuls arêtes de $\tilde{G}_{\bar{S}}$ dont le poids a une influence sur le poids total de P sont les arêtes de E^s et de E^{old} . Remarquons ensuite que l'ensemble des arêtes de G^+ est composé de l'union des arêtes de E^s et de E^{old} . Alors, P se traduit dans G^+ par un chemin composé uniquement d'arêtes dans $E^s \cup E^{\text{old}}$, i.e. d'arêtes de poids un. Le nombre de ces arêtes est donc exactement le poids de P qui est d'au plus d . Ainsi, il existe un chemin dans G' entre x et y de longueur au plus d .

Ensuite, il nous reste à vérifier que le nombre d'arêtes ajoutées dans G^+ soit d'au plus k par face de G . Ce fait découle du point (1) de la (d, k, s, w_J) -arête colorabilité de $E(\tilde{G}_{\bar{S}}) \setminus E(G)$, puisque les arêtes ajoutées par face sont les arêtes de E^s .

2. Réciproquement, supposons qu'il existe une complétion plane G^+ de G de diamètre au plus d et soit $\tilde{G}_{\bar{S}}$, une augmentation radiale de G pour un $\bar{S} \in \mathbf{S}_G^{(k)}$. Alors, si pour chaque couche de chaque grille annulaire de $\tilde{G}_{\bar{S}}$ nous appliquons l'Algorithme 3, nous obtenons une 3-partition $\mathbf{p} = \{E^0, E^s, E^\infty\}$ des arêtes de $E(\tilde{G}_{\bar{S}}) \setminus E(G)$ telle que E^0 (resp. E^s et E^∞) soit l'union des E_ℓ^0 de chaque grille annulaire de $\tilde{G}_{\bar{S}}$, pour tout $\ell \in [k]$. Alors, cette partition vérifie les trois propriétés de la (d, k, s, w) -arête colorabilité.

En effet, l'algorithme ne crée aucun chemin entre deux sommets de V^{old} , composé uniquement d'arêtes de E^0 , puisqu'il y force la présence d'une arête de E^s .

De plus, comme G^+ est une complétion plane à diamètre au plus d de G , il existe un chemin de longueur au plus d entre toute paire de sommets de V^{old} . Ce chemin emprunte soit des arêtes de $E(G)$, soit des arêtes ajoutées. Chaque arête ajoutées dans une face est simulée par un chemin d'arêtes de E^0 , d'aucune arête de E^∞ et de exactement une arête de E^s dans la grille annulaire de cette face. Ce qui donne un chemin de poids un. Et chaque arête de $E(G)$ a un poids de un. Enfin, et toujours parce que G^+ est une complétion plane à diamètre au plus d de G , au plus k arêtes ont été ajoutées dans G^+ . Donc chaque grille annulaire possède au plus k arêtes de E^s , puisque l'algorithme pour chaque arête ajoutée de G^+ crée une unique arête de E^s et n'en crée jamais autrement. ■

Programmation dynamique sur la (d, k) -arête colorabilité

Pour résoudre le problème pour le graphe G , un algorithme naïf fonctionnerait en générant toutes les 3-partitions valides des arêtes de $\tilde{G}_{\bar{s}}$ jusqu'à ce qu'une soit trouvée permettant au graphe d'admettre un diamètre borné par d . Cependant, la complexité d'un tel algorithme est exponentielle en la taille de G' . Ainsi, nous allons décrire un algorithme FPT basé sur une programmation dynamique sur une décomposition en coupes sphériques de $\tilde{G}_{\bar{s}}$.

Plus précisément, dans ce chapitre, nous allons prouver le lemme suivant.

Lemme 5

Il existe un algorithme qui, étant donné une entrée $(G, H, k, d, s, \mathbf{w})$, où G et H sont deux graphes plans tels que $G \leq H$ et H est 2-connexe, $k, d, s \in \mathbb{N}$, et $\mathbf{w} : E(G) \rightarrow \mathbb{N}$, et une décomposition en coupes sphériques (T, μ, π) de largeur au plus $b \in \mathbb{N}$, décide si $E(H) \setminus E(G)$ est (k, d, s, \mathbf{w}) -arête colorable dans H , et ce en $2^{O(b \cdot (\log b + \log d + \log k))}$ étapes.

3.1 Préliminaires

Nous utiliserons toujours les notations

- $E^{\text{old}} = E(G)$, $V^{\text{old}} = V(G)$,
- $E^{\text{new}} = E(H) \setminus E(G)$, $V^{\text{new}} = V(H) \setminus V(G)$.

3.1.1 Décomposition en coupes sphériques

Pour notre algorithme, nous construirons tout d'abord (T, μ, π) , une décomposition en coupes sphériques de H et de largeur b . Ensuite, nous choisirons arbitrairement une arête $e^* \in E(T)$ en la subdivisant pour ajouter un nouveau sommet v_{new} , puis nous mettrons à jour T en lui ajoutant un nouveau sommet r adjacent à v_{new} . Nous enracinons alors T en r et étendrons la fonction μ en posant $\mu(r) = \emptyset$.

Dans T nous appellerons *arête feuillue* une arête incidente à une feuille de T à l'exception de l'arête $e_r = \{r, v_{\text{new}}\}$. Une arête de T qui n'est pas feuillue sera appelée *interne*. Nous noterons $L(T)$ l'ensemble des arêtes feuillues de T et $I(T)$ l'ensemble de ses arêtes internes. Nous appellerons de même e_r *l'arête racinaire*.

Pour chaque $e \in E(T)$, soit T_e l'arbre de la forêt $T - e$ ne contenant pas le sommet r pour feuille et soit E_e les arêtes de H qui sont images par μ des feuilles de T_e . Posons alors $G_e = G[E_e]$ et $V_e = V(G_e)$ et observons que $G_{e_r} = G$. Pour chaque arête $e \in I(T)$, nous définissons comme l'ensemble de ses *fls* les deux arêtes partageant une extrémité commune avec e et se trouvant dans la composante connexe de $T - e$ ne contenant pas la racine r .

De plus, pour chaque arête $e \in E(T)$, Δ_e sera le disque fermé borné par O_e tel que $G \cap \Delta_e = G_e$. Enfin, pour chaque arête $e \in E(T)$, nous posons $V_e^{\text{new}} = V_e \cap V^{\text{new}}$, $V_e^{\text{old}} = V_e \cap V^{\text{old}}$, $E_e^{\text{new}} = E_e \cap E^{\text{new}}$, et $E_e^{\text{old}} = E_e \cap E^{\text{old}}$.

Nous posons $X_e = V_{O_e}$ et supposons toujours que les sommets de X_e sont énumérés par $\pi(e)$. Nous dénotons, de plus, par $\mathbf{A}_e = \{a_{1,2}, a_{2,3}, \dots, a_{k-1,k}, a_{k,1}\}$, l'ensemble des arcs de O_e , si $\pi(e) = [a_1, \dots, a_k, a_1]$, où la frontière de l'arc $a_{i,i+1}$ est composée des sommets a_i et a_{i+1} . Puis nous définissons $G_e^+ = (V_e, E(G_e \cup \mathbf{A}_e))$, i.e., G_e^+ est le graphe plan résultant si nous ajoutons comme nouvelles arêtes à G_e les arcs de O_e . Enfin, une face de G_e^+ est appelée *interne* lorsqu'elle n'est incidente à aucun arc de \mathbf{A}_e , i.e. si c'est aussi une face de G , et *marginale* si elle est strictement incluse dans une face de G .

3.1.2 (d, k) -configurations

Définition 16 ((d, k)-configuration d'un lasso)

Soit le lasso O_e . Une (d, k)-configuration de X_e est la donnée d'un quadruplet $((\mathcal{X}, \chi), (\mathcal{A}, \alpha), \mathcal{F}, \delta)$ tel que :

- (\mathcal{X}, χ) est une $[0, 1]$ -partition de X_e ,
- (\mathcal{A}, α) est une $[0, k]$ -partition de \mathbf{A}_e ,
- $\mathcal{F} \subseteq \{0, \dots, d\}^{X_e}$ et
- $\delta : X_e \times X_e \rightarrow \{0, \dots, d + 1\}$,

Intuitivement, chaque (d, k)-configuration d'un lasso va correspondre à une 3-partition des arêtes du sous-graphe induit par l'intérieur du lasso. Une (d, k)-configuration représentera alors une 3-partition valide dans le sens de la (d, k, s, w)-arête colorabilité.

Cependant, alors que le nombre des 3-partitions dépend de n , la taille des (d, k)-configurations ne dépendra que de d et de k .

Enfin, si nous voulons adapter les (d, k)-configurations au problème BCP, i.e. lorsque le nombre total d'arêtes ajoutées est d'au plus k , il nous suffira de remplacer les $[0, k]$ -partitions par un simple compte du nombre d'arêtes de E^s du sous-graphe induit par l'intérieur du lasso.

 $[t_1, t_2]$ -partitions

Soient X un ensemble et $t_1, t_2 \in \mathbb{N}$, où $t_1 \leq t_2$. La paire (\mathcal{X}, χ) est une $[t_1, t_2]$ -partition de X si \mathcal{X} est une collection de sous-ensembles non vides et deux-à-deux disjoints de X , et $\chi : \{1, \dots, |\mathcal{X}|\} \rightarrow \{t_1, \dots, t_2\}$ est une fonction.

Notons que dans le cas où $X = \emptyset$, une $[t_1, t_2]$ -partition correspond à la donnée de la paire $\{\emptyset, \emptyset\}$, où \emptyset dénote la fonction "vide", i.e. dont le domaine est vide.

Les partitions serviront à garantir les propriétés (1) et (3) de la (d, k, s, w)-arête colorabilité.

Fusion, restriction et intersection

Soient (\mathcal{X}_1, χ_1) et (\mathcal{X}_2, χ_2) deux $[t_1, t_2]$ -partitions respectivement des ensembles X_1 et X_2 telles que $\mathcal{X}_i = \{X_1^i, \dots, X_{\rho_i}^i\}$, $i \in \{1, 2\}$. Définissons $\mathcal{X}_1 \oplus \mathcal{X}_2$ comme suit. Si $x, x' \in X_1 \cup X_2$, alors nous dirons que $x \sim x'$ si et seulement si il existe un ensemble dans $\mathcal{X}_1 \cup \mathcal{X}_2$ contenant x et x' .

Soit \sim_T la fermeture transitive de \sim . Alors $\mathcal{X}_1 \oplus \mathcal{X}_2$ contient les classes d'équivalence de \sim_T . Définissons $\chi_1 \oplus \chi_2$ comme suit. Soit $\mathcal{X}_1 \oplus \mathcal{X}_2 = \{Y_1, \dots, Y_\rho\}$, alors pour chaque $i \in \{1, \dots, \rho\}$, nous posons

$$\chi_1 \oplus \chi_2(i) = \min\{t_2, \sum_{X_1^i \subseteq Y_i} \chi_1(i') + \sum_{X_2^i \subseteq Y_i} \chi_2(i')\}.$$

Alors la *fusion* des deux $[t_1, t_2]$ -partitions (\mathcal{X}_1, χ_1) et (\mathcal{X}_2, χ_2) est la donnée de la paire $(\mathcal{X}_1 \oplus \mathcal{X}_2, \chi_1 \oplus \chi_2)$ qui est aussi une $[t_1, t_2]$ -partition, notée $(\mathcal{X}_1, \chi_1) \oplus (\mathcal{X}_2, \chi_2)$.

Soient (\mathcal{X}, χ) une $[t_1, t_2]$ -partition de l'ensemble X et un sous-ensemble $X' \subseteq X$. Nous définissons la *restriction* de (\mathcal{X}, χ) à X' , notée $(\mathcal{X}, \chi)|_{X'}$, comme la $[t_1, t_2]$ -partition (\mathcal{X}', χ') de X' telle que $\mathcal{X}' = \{X_i \cap X' \mid X_i \in \mathcal{X}\} \setminus \{\emptyset\}$ et $\chi' = \{(i, \chi(i)) \mid X_i \cap X' \neq \emptyset\}$. Posons de même que l'*intersection* de (\mathcal{X}, χ) avec X' sera la $[t_1, t_2]$ -partition (\mathcal{X}', χ') , notée $(\mathcal{X}, \chi) \cap X'$, telle que $\mathcal{X}' = \{X_i \in \mathcal{X} \mid X_i \cap (X \setminus X') \neq \emptyset\}$, $\chi' = \{(i, \chi(i)) \mid X_i \cap X'' \neq \emptyset\}$ et $\chi' = \{(i, \chi(i)) \mid X_i \cap X'' \neq \emptyset\}$, où $X'' = \cup_{X'_i \in \mathcal{X}'} X'_i$.

Remarquons enfin que $(\mathcal{X}, \chi)|_{X'}$ et $(\mathcal{X}, \chi) \cap X'$ ne sont pas toujours égaux.

Distances

Étant donnée une arête $e \in E(T)$ et une 3-partition $\mathbf{p} = \{E_e^0, E_e^s, E_e^\infty\}$ des arêtes de $G_e \setminus E(G \cap G_e)$. Posons $\mathbf{w}_\mathbf{p} = \mathbf{w} \cup \{(e, 0) \mid E \in E^0\} \cup \{(e, s) \mid E \in E^s\} \cup \{(e, d+1) \mid E \in E^\infty\}$. Alors nous avons les deux définitions suivantes.

Définition 17 (*Distance relative à une partition*)

Soient $x, x' \in V(G_e)$, alors la distance entre x et x' relativement à \mathbf{p} est donnée par

$$\mathbf{w}_\mathbf{p}\text{-dist}_G(x, x') = \min\{\mathbf{w}_\mathbf{p}(E(P)) \mid P \text{ est un } (x, x')\text{-chemin dans } G\}.$$

Définition 18 (*X_e -signature*)

Pour chaque sommet $v \in V_e$, nous définissons sa X_e -signature comme la fonction $\phi_v : X_e \rightarrow \{0, \dots, d\}$ telle que si $x \in X_e$, alors $\phi_v(x) = \min\{\mathbf{w}_\mathbf{p}\text{-dist}_{G_e}(v, x), d+1\}$.

Les fonctions de distance serviront quant à elles à garantir la propriété (2) de la (d, k, s, \mathbf{w}) -arête colorabilité.

3.2 Programmation dynamique

3.2.1 Les tables

Nous devons alors expliciter une programmation dynamique sur l'arbre T de la décomposition en coupes sphériques (T, μ, π) . Dans ce but, nous avons besoin de décrire, pour chaque arête $e \in E(T)$, une table contenant l'information sur les solutions partielles du problème restreint au graphe G_e et ce de manière à ce que la table de l'arête $e \in E(T)$ soit calculable à partir des tables des ses deux fils. Notons que la taille de chaque table ne dépendra pas de $|G|$ et que la solution finale sera dérivée de la table de l'arête racinaire e_r .

La fonction \mathfrak{T}

Nous définissons alors la fonction \mathfrak{T} faisant correspondre chaque $e \in E(T)$ à une collection $\mathfrak{T}(e)$ de $(d, k + 1)$ -configurations de X_e . Et nous dirons que

$$Q = ((\mathcal{X}, \chi), (\mathcal{A}, \alpha), (\mathcal{F}, \mathcal{E}), \delta) \in \mathfrak{T}(e)$$

si il existe une 3-partition $\mathbf{p} = \{E_e^0, E_e^s, E_e^\infty\}$ de E_e^{new} telle que les propriétés suivantes sont vérifiées.

- (a) Si C_1, \dots, C_h sont les composantes connexes de (V_e, E_e^0) , alors :
 - $\mathcal{X} = \{V(C_1) \cap X_e, \dots, V(C_h) \cap X_e\}$ et
 - $\forall_{j \in \{1, \dots, h\}} \chi(i) = 0$ si C_i ne contient aucun sommet de V_e^{old} , sinon $\chi(i) = 1$.
- (b) Chaque chemin dans G_e entre deux sommets de V_e^{old} contient au moins une arête de $E_e^s \cup E_e^\infty \cup E_e^{\text{old}}$.
- (c) \mathcal{A} est une partition de \mathbf{A}_e telle que deux arcs $A, A' \in \mathbf{A}_e$ appartiennent au même ensemble A_i de \mathcal{A} si et seulement si ils sont incidents à la même face marginale F_i de G_e^+ . De plus, pour chaque $i \in [|A|]$, $\alpha(i)$ est égal au nombre d'arêtes de E_e^s qui sont dans l'intérieur de F_i .
- (d) Toute face interne de G_e^+ contient au plus k arêtes de E_e^s .
- (e) $(\mathcal{F}, \mathcal{E})$ est un graphe tel que $\mathcal{F} = \{\phi_v \mid v \in V_e\}$.
- (f) Une arête $\{\phi, \phi'\}$ appartient à \mathcal{E} si et seulement si il existe deux sommets $v, v' \in V_e$, où $\mathbf{w}_p\text{-dist}_{G_e}(v, v') > d$ tel que $\phi = \phi_v$ et $\phi' = \phi_{v'}$.
- (g) Pour toute paire de sommets $v, v' \in V_e$, nous avons soit $\mathbf{w}_p\text{-dist}_{G_e}(v, v') \leq d$, soit il existe deux sommets $x, x' \in X_e$ tels que $\phi_v(x) + \phi_{v'}(x') \leq d$.
- (h) Pour toute paire $x, x' \in X_e$, $\delta(x, x') = \mathbf{w}_p\text{-dist}_{G_e}(x, x')$.

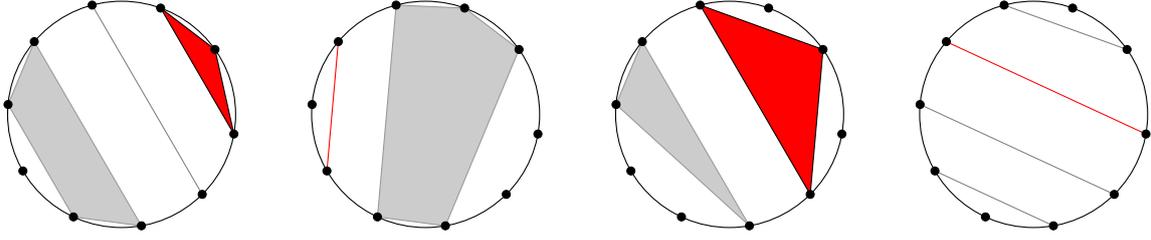
Notons que si nous voulons résoudre le problème BCP, i.e. lorsque le nombre total d'arêtes ajoutées est d'au plus k , il nous suffira de supprimer la propriété (c) et de remplacer la propriété (d) par la propriété :

(d.bis) G_e contient au plus k arêtes de E_e^s .

La fonction $\mathfrak{T}(e)$ et la (k, d, s, w) -arête colorabilité

Nous allons justifier les propriétés précédentes.

- (a) et (b) Lorsque nous contracterons les arêtes de E_e^0 , tous les sommets des composantes de G_e formées par les arêtes de E_e^0 seront toutes réduites à un unique sommet. Il nous faut alors interdire que dans une telle composante se trouvent deux sommets différents de V_e^{old} . Nous allons alors marquer chaque telle composante, stockée par sa partitions des sommets de X_e correspondante, avec comme couleur le nombre de sommets de V_e^{old} qu'elle contient. Si cette couleur dépasse alors un, la 3-partition $\{E_e^0, E_e^s, E_e^\infty\}$ des arêtes de G_e doit être interdite, voir Figure 3.1.

FIGURE 3.1 – $[0, 1]$ -partitions des sommets d'un lasso.

- (c) et (d) Une des contraintes principales du problème BFCP est le fait qu'on ne puisse ajouter qu'au plus k arêtes par face. Si donc dans un graphe G_e , il existe une face interne ou marginale possédant plus de k arêtes de E_e^s , la 3-partition correspondante doit être interdite. De plus, et afin de garder trace du nombre d'arêtes ajoutées dans chaque face de G_e , nous stockons le nombre d'arêtes de E_e^s des faces marginales, puisque lorsque nous procéderons à la fusion de deux tables, seules les faces marginales pourront voir leur nombre d'arêtes augmenter.
- (e),(f) et (g) Lorsque dans G_e deux sommets x et y sont éloignés, i.e. à une distance strictement supérieure à d , alors le seul moyen pour que la 3-partition des arêtes de G_e reste valide est qu'il existe un chemin de x et un autre de y vers le lasso, tous deux d'une taille au plus égale à d . En effet, ainsi il est potentiellement possible de joindre x et y par un chemin empruntant des arêtes d'autres lasso. Dans ce cas, nous dirons que x et y sont *potentiellement proches*, voir Figure 3.2. C'est pourquoi, nous conservons dans \mathcal{F} toutes les possibilités de joindre un sommet de G_e avec un sommet de son lasso. Pour chaque possibilité, nous conservons la distance si elle est inférieure ou égale à d et le sommet de X_e correspondant, i.e. nous conservons les toutes les X_e -signatures. Bien sûr, il est possible que deux sommets de G_e aient la même X_e -signature. Le nombre des tels signatures est en réalité borné par une fonction de $|X_e|$ et de d . Nous stockons alors ces informations sous la forme d'un graphe dont les sommets sont les X_e -signatures et qui sont adjacentes s'il existe deux sommets de G_e éloignés mais potentiellement proches réalisant ces signatures.
- (h) Nous stockons pour chaque lasso O_e les distances entre deux sommets de X_e . C'est un sous-ensemble de \mathcal{F} mais avoir accès à cette information immédiatement nous sera utile par la suite, voir Figure 3.3.

La proposition suivante justifie le lien entre la fonction $\mathfrak{T}(e)$ et la (k, d, s, \mathbf{w}) -arête colorabilité.

Proposition 14

E_{new} est (k, d, s, \mathbf{w}) -arête colorable dans H si et seulement si $\mathfrak{T}(e_r) \neq \emptyset$, i.e. $\mathfrak{T}(e_r) = \{((\emptyset, \emptyset), (\emptyset, \emptyset), \emptyset, \emptyset)\}$.

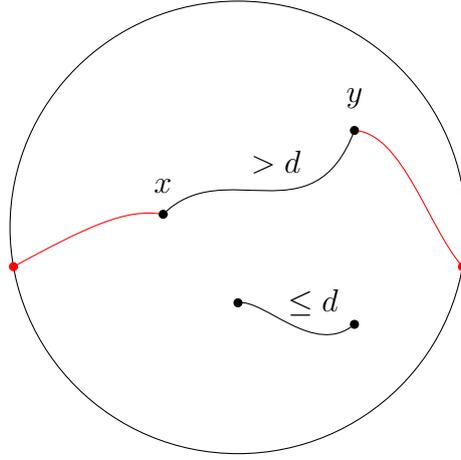


FIGURE 3.2 – La table stockée dépend de $|X_e|$ et de d uniquement.

Démonstration : Il existe une 3-partition $\{E^0, E^s, E^\infty\}$ de E_{new} vérifiant les propriétés de la (k, d, s, w) -arête colorabilité dans H si et seulement si

- aucun sommet de H n'est à une distance de plus de d ,
- il n'existe aucune composante de H induite par les arêtes de E^0 contenant plus d'un sommet de V^{old} et
- toute face de G possède au plus k arêtes dans E^s .

Comme $\mathfrak{T}(e_r)$ représente la solution partielle associée au graphe tout entier, i.e. la solution finale, et que toutes ses composantes sont vides, cela signifie que la 3-partition de H est valide. ■

Tables(e)

Nous allons maintenant donner l'ensemble des tables de notre programmation dynamique pour chaque $e \in T(E)$. En particulier, nous montrerons comment calculer une fonction **tables** qui fait correspondre chaque arête $e \in E(T)$ à une collection de $(d, k + 1)$ -configurations de X_e telle que $\mathfrak{T}(e) = \mathbf{tables}(e)$.

Pour le calcul de **tables**, nous utiliserons une programmation dynamique basée sur un traitement des arêtes de T du bas vers le haut. Notre première étape sera de donner **tables**(e_l) pour chaque $e_l \in L(T)$ puis de donner une procédure, appelée **jointure**, décrite dans l'Algorithme 4 et qui à partir d'une arête $e \in I(T)$ et des ses deux fils e_l et e_r et leurs tables **tables**(e_l) et **tables**(e_r) en entrées, retournera la table **tables**(e).

La définition de **tables** que nous allons donner garantira les deux propriétés suivantes.

- Si e_r est une arête feuillue, alors $\mathfrak{T}(e_l) = \mathbf{tables}(e_l)$.
- Si pour une arête $e \in E(T)$, dont les deux fils sont e_l et e_r , nous avons $\mathfrak{T}(e_l) = \mathbf{tables}(e_l)$ et $\mathfrak{T}(e_r) = \mathbf{tables}(e_r)$. Alors $\mathfrak{T}(e) = \mathbf{tables}(e)$.

Par la Proposition 14, nous voyons qu'il est suffisant de donner une procédure permettant de calculer $\mathfrak{T}(e)$ pour tout $e \in E(T)$. Nous allons détailler les tables d'une arête feuillue $e \in L(T)$ puis nous donnerons une procédure pour les arêtes internes $e \in I(T)$.

Supposons maintenant que $e_l \in L(T)$ telle que $\pi(e_l) = [a_1, a_2, a_1]$ et $\mathbf{A}_{e_l} = \{a_{1,2}, a_{2,1}\}$.

- Si $\{a_1, a_2\} \in E_e^{\text{old}}$, alors

$$\begin{aligned} \text{tables}(e_l) = & \{ ((\{a_1\}, \{a_2\}), \{(1, 1), (2, 1)\}), \\ & (\{a_{1,2}\}, \{a_{2,1}\}), \{(1, 0), (2, 0)\}), \\ & (\{(a_1, 0), (a_2, \mathbf{w}(\{a_1, a_2\}))\}, \{(a_1, \mathbf{w}(\{a_1, a_2\})), (a_2, 0)\}, K)\}, \\ & \{((a_1, a_2), \mathbf{w}(\{a_1, a_2\}))\} \}, \end{aligned}$$

où K est sans boucle et, de plus, $K = \emptyset$ si et seulement si $\mathbf{w}(\{a_1, a_2\}) \leq d$.

- Si $\{a_1, a_2\} \in E_e^{\text{new}}$ et $\{a_1, a_2\} \subseteq V_e^{\text{old}}$, alors $\text{tables}(e_l) = \{Q^s, Q^\infty\}$, où

$$\begin{aligned} Q^s = & ((\{a_1\}, \{a_2\}), \{(1, \langle a_1 \in V_e^{\text{old}} \rangle), (2, \langle a_2 \in V_e^{\text{old}} \rangle)\}) \\ & (\{a_{1,2}, a_{2,1}\}, \{(1, 1)\}) \\ & (\{(a_1, s), (a_2, s)\}, K) \\ & \{((a_1, a_2), s)\}, \end{aligned}$$

où K est sans boucle et, de plus, $K = \emptyset$ si et seulement si $s \leq d$,

$$\begin{aligned} Q^\infty = & ((\{a_1\}, \{a_2\}), \{(1, \langle a_1 \in V_e^{\text{old}} \rangle), (2, \langle a_2 \in V_e^{\text{old}} \rangle)\}) \\ & (\{a_{1,2}, a_{2,1}\}, \{(1, 0)\}) \\ & (\{(a_1, d+1), (a_2, d+1)\}, K) \\ & \{((a_1, a_2), d+1)\}, \end{aligned}$$

où K contient une unique arête qui n'est pas une boucle.

- Si $\{a_1, a_2\} \in E_e^{\text{new}}$ and $V_e^{\text{new}} \cap \{a_1, a_2\} \neq \emptyset$, alors $\text{tables}(e_l) = \{Q^0, Q^s, Q^\infty\}$, où

$$\begin{aligned} Q^0 = & ((\{a_1, a_2\}), \{(1, 1 - \langle \{a_1, a_2\} \subseteq V_e^{\text{new}} \rangle)\}) \\ & (\{a_{1,2}, a_{2,1}\}, \{(1, 0)\}) \\ & (\{(a_1, 0), (a_2, 0)\}, \emptyset) \\ & \{((a_1, a_2), 0)\}. \end{aligned}$$

3.2.2 Jointure et fusion

Les algorithmes

Nous allons maintenant décrire la procédure permettant de joindre les deux tables des fils d'une arête interne. Remarquons tout d'abord que X_e (resp. \mathbf{A}_e) est la différence symétrique de X_{e_l} et de X_{e_r} (resp. \mathbf{A}_{e_l} et \mathbf{A}_{e_r}). Posons ensuite que $X_e^F = (X_{e_l} \cup X_{e_r}) \setminus X_e$. Alors la procédure **jointure** est décrite dans l'Algorithme 4.

Il nous reste alors à décrire la routine **fusion** permettant de calculer la (d, k) -configuration $((\mathcal{X}, \chi), (\mathcal{A}, \alpha), (\mathcal{F}, \mathcal{E}), \delta)$ résultant de la fusion des deux (d, k) -configurations $((\mathcal{X}_l, \chi_l), (\mathcal{A}_l, \alpha_l), (\mathcal{F}_l, \mathcal{E}_l), \delta_l)$ et $((\mathcal{X}_r, \chi_r), (\mathcal{A}_r, \alpha_r), (\mathcal{F}_r, \mathcal{E}_r), \delta_r)$. La procédure **fusion** est donnée par l'Algorithme 5.

Entrées : Deux collections \mathcal{C}_l et \mathcal{C}_r de (d, k) -configurations de X_{e_l} et de X_{e_r} .

Output : Une collection \mathcal{C} de (d, k) -configurations de X_e .

```

début
   $\mathcal{C} = \emptyset$ 
  pour  $(Q_l, Q_r) \in \mathcal{C}_l \times \mathcal{C}_r$  faire
    si  $\text{fusion}(Q_l, Q_r) \neq \text{void}$  alors
       $\mathcal{C} \leftarrow \mathcal{C} \cup \{\text{fusion}(Q_l, Q_r)\}$ 
    fin
  fin
  retourner  $\mathcal{C}$ 
fin

```

Algorithme 4: jointure

Entrées : $((\mathcal{X}_l, \chi_l), (\mathcal{A}_l, \alpha_l), (\mathcal{F}_l, \mathcal{E}_l), \delta_l)$ et $((\mathcal{X}_r, \chi_r), (\mathcal{A}_r, \alpha_r), (\mathcal{F}_r, \mathcal{E}_r), \delta_r)$.

Output : La (d, k) -configuration $((\mathcal{X}, \chi), (\mathcal{A}, \alpha), (\mathcal{F}, \mathcal{E}), \delta)$.

```

début
   $(\mathcal{X}', \chi') = (\mathcal{X}_l, \chi_l) \oplus (\mathcal{X}_r, \chi_r)$ 
  si  $\chi'^{-1}(2) \neq \emptyset$  alors
    retourner void
  fin
   $(\mathcal{X}, \chi) = (\mathcal{X}_l, \chi_l) \oplus (\mathcal{X}_r, \chi_r)|_{V_e}$ 
   $(\mathcal{A}', \alpha') = (\mathcal{A}_l, \alpha_l) \oplus (\mathcal{A}_r, \alpha_r)$ 
  si  $\alpha'^{-1}(k+1) \neq \emptyset$  alors
    retourner void
  fin
   $(\mathcal{A}, \alpha) = (\mathcal{A}_l, \alpha_l) \oplus (\mathcal{A}_r, \alpha_r)|_{\mathbf{A}_e}$ 
  Calculer la fonction  $\gamma : (\mathcal{F}_l \cup \mathcal{F}_r \cup X_e) \times (\mathcal{F}_l \cup \mathcal{F}_r \cup X_e) \rightarrow [d]$ 
  Calculer le nouveau graphe  $\mathcal{G} = (\mathcal{F}, \mathcal{E})$ 
  si  $\mathcal{G}$  est le graphe nul alors
    retourner void
  fin
   $\delta = \{((x, x'), \gamma(x, x')) \mid x, x' \in X_e\}$ 
  retourner  $((\mathcal{X}, \chi), (\mathcal{A}, \alpha), (\mathcal{F}, \mathcal{E}), \delta)$ 
fin

```

Algorithme 5: fusion

Calcul de la fonction γ

Une illustration du calcul de la fonction γ est donnée dans la Figure 3.3. Ce calcul s'effectue comme suit. Soient δ_l et δ_r , nous définissons

$$\gamma : (\mathcal{F}_{e_l} \cup \mathcal{F}_{e_r} \cup X_e) \times (\mathcal{F}_{e_l} \cup \mathcal{F}_{e_r} \cup X_e) \rightarrow [d]$$

en distinguant plusieurs cas.

- i. Si $(x \in X_e \setminus X_{e_r} \wedge \phi \in \mathcal{F}_l)$ ou $(x \in X_e \setminus X_{e_l} \wedge \phi \in \mathcal{F}_r)$, alors

$$\begin{aligned} \gamma(\phi, x) = \min \{ & \phi(x), \min \{ \phi(p_1) + \sum_{[\rho-1]} \delta_{\mathbf{s}(i)}(p_i, p_{i+1}) + \\ & \delta_{\mathbf{s}(\rho)}(p_\rho, x) \mid [p_1, \dots, p_\rho] \in \text{ord}^0(X_e^F) \} \}, \end{aligned}$$

où $\mathbf{s}(i) = \text{"l"}$ si $\langle x \in X_e \setminus X_{e_l} \rangle = (i \bmod 2)$, sinon $\mathbf{s}(i) = \text{"r"}$.

- ii. Si $(x \in X_e \setminus X_{e_l} \wedge \phi \in \mathcal{F}_l)$ or $(x \in X_e \setminus X_{e_r} \wedge \phi \in \mathcal{F}_r)$, alors

$$\gamma(\phi, x) = \min \{ \phi(p_1) + \sum_{[\rho-1]} \delta_{\mathbf{t}(i)}(p_i, p_{i+1}) + \delta_{\mathbf{t}(\rho)}(p_\rho, x) \mid [p_1, \dots, p_\rho] \in \text{ord}^1(X_e^F) \},$$

où $\mathbf{t}(i) = \text{"l"}$ si $\langle x \in X_e \setminus X_{e_l} \rangle \neq (i \bmod 2)$, sinon $\mathbf{t}(i) = \text{"r"}$.

- iii. Si x est un des (au plus deux) sommets de $(X_{e_r} \cap X_{e_l}) \setminus X_e^F$ and $\phi \in \mathcal{F}_l \cup \mathcal{F}_r$, alors

$$\begin{aligned} \gamma(\phi, x) = \min \{ & \phi(x), \\ & \min \{ \phi(p_1) + \sum_{[\rho-1]} \delta_{\mathbf{u}(i)}(p_i, p_{i+1}) + \\ & \delta_{\mathbf{u}(q)}(p_\rho, x) \mid [p_1 \dots, p_\rho] \in \text{ord}^q(X_e^F) \} \mid q \in \{0, 1\} \}, \end{aligned}$$

où $\mathbf{u}(i) = \text{"r"}$ si $\langle \phi \in \mathcal{F}_l \rangle = (i \bmod 2)$, sinon $\mathbf{u}(i) = \text{"l"}$.

- iv. Si $\phi, \phi' \in \mathcal{F}_l \cup \mathcal{F}_r$, alors

$$\gamma(\phi, \phi') = \min \{ \phi(p_1) + \sum_{[\rho-1]} \delta_{\mathbf{u}(i)}(p_i, p_{i+1}) + \phi'(p_\rho) \mid [p_1, \dots, p_\rho] \in \text{ord}^q(X_e^F) \}.$$

Dans cette égalité, $q = 1$ si ϕ et ϕ' se trouvent dans deux ensembles différents de $\{\mathcal{F}_l, \mathcal{F}_r\}$, sinon $q = 0$. La fonction \mathbf{u} est identique à celle du cas précédent.

v. Si $x_1, x_2 \in X_e \setminus X_{e_r}$ or $x_1, x_2 \in X_e \setminus X_{e_l}$, alors

$$\delta(x_1, x_2) = \min \left\{ \delta_{\mathbf{y}(0, x_1)}(x_1, x_2), \min \left\{ \delta_{\mathbf{y}(0, x_1)}(x_1, p_1) + \sum_{i \in [\rho-1]} \delta_{\mathbf{y}(i, x_1)}(p_i, p_{i+1}) + \delta_{\mathbf{y}(0, x_2)}(p_\rho, x_2) \mid [p_1, \dots, p_\rho] \in \text{ord}^0(X_e^F) \right\} \right\}.$$

Dans cette égalité, $\mathbf{y}(i, x) = \text{"l"}$ si $\langle x \in X_e \setminus X_{e_r} \rangle = \langle i \bmod 2 = 0 \rangle$, sinon $\mathbf{y}(i, x) = \text{"r"}$.

vi. Si x_1, x_2 se trouvent dans deux ensembles différents de $\{X_e \setminus X_{e_r}, X_e \setminus X_{e_l}\}$, alors

$$\delta(x_1, x_2) = \min \left\{ \delta_{\mathbf{y}(0, x_1)}(x_1, p_1) + \sum_{i \in [\rho-1]} \delta_{\mathbf{y}(i, x_1)}(p_i, p_{i+1}) + \delta_{\mathbf{y}(0, x_2)}(p_\rho, x_2) \mid [p_1, \dots, p_\rho] \in \text{ord}^1(X_e^F) \right\}.$$

La fonction \mathbf{y} est identique à celle du cas précédent.

vii. Si exactement un de x_1, x_2 , disons x_2 , se trouve dans $X_{e_r} \cap X_{e_l} \setminus X_e^F$, alors

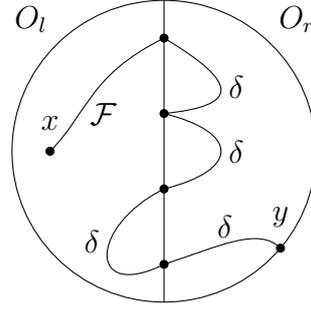
$$\delta(x_1, x_2) = \min \left\{ \delta_{\mathbf{y}(0, x_1)}(x_1, x_2), \min \left\{ \min \left\{ \delta_{\mathbf{y}(0, x_1)}(x_1, p_1) + \sum_{i \in [\rho-1]} \delta_{\mathbf{y}(i, x_1)}(p_i, p_{i+1}) + \delta_{\mathbf{y}(0, x_2)}(p_\rho, x_2) \mid [p_1, \dots, p_\rho] \in \text{ord}^q(X_e^F) \right\} \mid q \in \{0, 1\} \right\} \right\}.$$

La fonction \mathbf{y} est identique à celle du cas précédent. Et dans le cas où c'est x_1 et non x_2 qui se trouve dans $(X_{e_r} \cap X_{e_l}) \setminus X_e^F$, alors il suffira d'échanger leurs positions dans l'équation précédente.

viii. Si à la fois x_1 et x_2 se trouvent dans $X_{e_r} \cap X_{e_l} \setminus X_e^F$, alors

$$\delta(x_1, x_2) = \min \left\{ \delta_l(x_1, x_2), \delta_r(x_1, x_2), \min \left\{ \min \left\{ \delta_{\mathbf{z}(0, j)}(x_1, p_1) + \sum_{i \in [\rho-1]} \delta_{\mathbf{z}(i, j)}(p_i, p_{i+1}) + \delta_{\mathbf{z}(q, j)}(p_\rho, x_2) \mid [p_1, \dots, p_\rho] \in \text{ord}^q(X_e^F) \right\} \mid (q, j) \in \{0, 1\}^2 \right\} \right\}.$$

Dans cette égalité, $\mathbf{z}(i, j) = \text{"l"}$ si $(i + j \bmod 2) = 0$, sinon $\mathbf{z}(i, j) = \text{"r"}$.

FIGURE 3.3 – Nouvelle distance entre x et y induite par la fusion des lassos O_l et O_r

Calcul du graphe $\mathcal{G} = (\mathcal{F}, \mathcal{E})$

Pour construire le graphe $\mathcal{G} = (\mathcal{F}, \mathcal{E})$ à partir des deux graphes $(\mathcal{F}_l, \mathcal{E}_l)$ et $(\mathcal{F}_r, \mathcal{E}_r)$, nous procédons comme suit.

1. Considérons le graphe résultant de l'union jointe des graphes $(\mathcal{F}_l, \mathcal{E}_l)$ and $(\mathcal{F}_r, \mathcal{E}_r)$ et supprimons lui toutes les arêtes $\{\phi_1, \phi_2\}$ pour lesquelles $\gamma(\phi_1, \phi_2) \leq d$. Et soit $\mathcal{G}^+ = (\mathcal{F}^+, \mathcal{E}^+)$ le graphe obtenu après ces suppressions.
2. Si pour une arête $\{\phi_1, \phi_2\} \in \mathcal{E}^+$ nous avons $x_1, x_2 \in V_e$, $\gamma(\phi_1, x_1) + \gamma(\phi_2, x_2) > d$, pour chaque $x_1, x_2 \in X_e$, alors nous transformons le graphe en un graphe nul.
3. Considérons la fonction $\lambda : \mathcal{F}_l \cup \mathcal{F}_r \rightarrow \{1, \dots, d\}^{X_e}$ telle que

$$\lambda(\phi) = \{(x, \gamma(\phi, x)) \mid x \in X_e\}$$

4. Pour tout $\phi' \in \lambda(\mathcal{F}_l \cup \mathcal{F}_r)$ et pour chaque ensemble $F = \lambda^{-1}(\phi')$, identifions dans \mathcal{G}^+ tous les sommets dans F et si au moins une paire d'entre eux est adjacente dans \mathcal{G}^+ , alors contractons l'arête induite par la paire, puis ajoutons une boucle sur le sommet créé par la contraction. Notons $\mathcal{G} = (\mathcal{F}, \mathcal{E})$ le graphe obtenu après ces contractions.

Remarquons que $\mathcal{F} = \lambda(\mathcal{F}_l \cup \mathcal{F}_r)$.

3.2.3 Complexité de la programmation dynamique

Donner la complexité de la programmation dynamique c'est tout d'abord donner la taille d'une table. La taille d'une table dépend uniquement du nombre de sommets de chaque lasso. Rappelons que nous avons posé que, pour toute arête $e \in E(T)$ de la décomposition en coupes sphériques, $|X_e| \leq b \in \mathbb{N}$. Ainsi, nous pouvons donner la complexité de la programmation dynamique uniquement en fonction de b .

Taille des tables

La proposition suivante nous permet de prouver que la taille des partitions dépend uniquement de la largeur arborescente.

Proposition 15 ([Kreweras, 1972])

Le nombre de partitions non-croisées d'un cycle de taille $b \in \mathbb{N}$ est le $b^{\text{ième}}$ nombre de Catalan :

$$C_b = \frac{1}{b+1} \binom{2b}{b} \sim 4^b = 2^{O(b)}.$$

Nous en déduisons donc le nombre de (d, k) -configurations.

Proposition 16

Soient un ensemble X , avec $|X| \leq b$. Alors il y a au plus $2^{O(b \cdot (\log k + \log d))}$ (d, k) -configurations de X .

Démonstration : Remarquons que les $[0, k]$ -partitions ou les $[0, 1]$ -partitions de X correspondent aux partitions non-croisées de X , où ce dernier est regardé comme un cycle, que nous aurions coloré par une couleur de l'ensemble $\{0, \dots, k\}$ ou $\{0, 1\}$. En effet, nous identifions chacun de leurs sous-ensembles de sommets avec les composantes connexes d'arêtes de E^0 ou des faces de G_e^+ , et bien entendu deux composantes connexes comme deux faces ne se croisent pas, et nous les colorons selon soit le nombre de sommets de V^{old} , soit le nombre d'arêtes de E^s qu'elles contiennent.

Sans perte de généralité, puisque seul les cas où $k \geq 1$ sont pertinents, nous pouvons restreindre notre analyse aux $[0, k]$ -partitions de X . Fixons une telle partition. Son nombre de sous-ensembles est borné par b , c'est le cas où chaque sous-ensemble est un singleton. Le nombre de façons possibles de colorer les éléments de cette partition est alors de $k^b = 2^{O(b \cdot \log k)}$.

De plus, par la Proposition 15, le nombre de telles partitions de X est d'au plus 4^b . Ainsi, le nombre de $[0, k]$ -partitions de X est de $4^b \cdot k^b = (4k)^b = 2^{O(b \cdot \log k)}$, et nous pouvons raisonnablement affirmer que le nombre de $[0, 1]$ -partitions de X est négligeable devant celui des $[0, k]$ -partitions.

Ensuite, le nombre de graphes possibles du type $(\mathcal{F}, \mathcal{E})$ est celui du nombre de fonctions de l'ensemble X vers l'ensemble $\{0, \dots, d\}$ qui est de $(d+1)^b = 2^{O(b \cdot \log d)}$.

Enfin, le nombre de δ possibles est la taille du produit cartésien entre l'ensemble des couples de X , i.e. $X \times X$ et l'ensemble $\{0, \dots, d+1\}$ qui est de $b^2 \cdot (d+2)$. Et nous pouvons encore raisonnablement affirmer que la taille de δ , qui est un sous-ensemble de \mathcal{F} , est négligeable devant celle de \mathcal{F} .

Ainsi, le nombre de (d, k) -configurations possibles de l'ensemble X est de

$$2^{O(b \cdot \log k)} \cdot 2^{O(b \cdot \log d)} = 2^{O(b \cdot (\log k + \log d))}.$$

■

Lorsque l'on restreint les (d, k) -configurations à leur définition adaptée au problème BCP, i.e. lorsque le nombre total d'arêtes ajoutées est d'au plus k , nous obtenons le corollaire suivant à la place de la Proposition 16.

Corollaire 5

Soient un ensemble X , avec $|X| \leq b$. Alors il y a au plus $2^{O(b \cdot \log d)}$ (d, k) -configurations de X pour le problème BCP.

Temps d'exécution des algorithmes

Rappelons que pour tout $e \in E(T)$ nous avons posé que $|V_e| = b \in \mathbb{N}$. Alors la proposition suivante nous donne la complexité de la procédure **fusion**.

Proposition 17

La procédure **fusion** décrite dans l'Algorithme 5 s'exécute en $2^{O(k+b \cdot (\log b + \log d))}$ étapes.

Démonstration : Seules deux étapes de cet algorithme sont non-triviales en terme de complexité. Dont en premier lieu, le calcul de la fonction γ . Remarquons que γ admet au plus $((d+1)^{|X_{e_l}|} + (d+1)^{|X_{e_r}|} + |X_e|)^2 = 2^{O(b \cdot \log d)}$ entrées et que chacune de leurs valeurs nécessite de parcourir toutes les permutations possibles des sous-ensembles de X_e^F , soit un nombre de $b! = 2^{O(b \cdot \log b)}$. Nous déduisons de ces deux remarques que le calcul de la fonction γ nécessite $2^{O(b \cdot (\log b + \log d))}$ étapes.

Remarquons enfin que le graphe $\mathcal{G} = (\mathcal{F}, \mathcal{E})$ possède $2^{O(b \cdot \log d)}$ sommets et que son calcul ne nécessite que des opérations d'ordre linéaire en son nombre de sommets. Ce qui est négligeable devant la complexité du calcul de la fonction γ .

Ainsi, l'exécution de la procédure **fusion** nécessite au plus $2^{O(b \cdot (\log b + \log d))}$ étapes. ■

Enfin, voici la complexité de la procédure **jointure**.

Proposition 18

La procédure **jointure** décrite dans l'Algorithme 4 s'exécute en $2^{O(b \cdot (\log b + \log d + \log k))}$ étapes.

Démonstration : Nous avons prouvé, dans la Proposition 16 que $|\mathfrak{T}(e)| = 2^{O(b \cdot (\log d + \log k))}$. Alors, la procédure **fusion** sera exécutée au plus $2^{O(b \cdot (\log d + \log k))}$ fois dans la procédure **jointure**. Et par la Proposition 17, nous savons qu'une exécution de la procédure **fusion** s'effectue en $2^{O(b \cdot (\log b + \log d))}$ étapes. Nous en déduisons que la procédure **jointure** s'exécute en temps

$$2^{O(b \cdot (\log b + \log d + \log k))}.$$

■

Remarquons encore une fois que lorsque nous nous restreignons au problème BCP, i.e. lorsque le nombre total d'arêtes ajoutées est d'au plus k , alors par le Corollaire 5, nous en déduisons facilement le corollaire suivant.

Corollaire 6

La procédure **jointure** décrite dans l'Algorithme 4 s'exécute en $2^{O(b \cdot (\log b + \log d))}$ étapes pour le problème BCP.

3.3 Un algorithme FPT pour le problème BCP

3.3.1 L'algorithme pour BFCP

Nous avons maintenant assez d'outils pour construire un algorithme permettant de résoudre le problème BFCP, dans lequel le nombre d'arêtes ajoutées par face est d'au plus k . En particulier, le Lemme 5 nous donne un algorithme permettant de décider si étant donné une entrée (G, J, H, d, k, s, w_J) , où G et H sont deux graphes plans tels que $G \leq H$ et H est 2-connexe, $J \subseteq E(G)$, $k, d, s \in \mathbb{N}$, et $w_J : E(G) \rightarrow \mathbb{N}$ telle que $w_J = \{(e, \langle e \notin J \rangle) \mid e \in E(G)\}$, et une décomposition en coupes sphériques (T, μ, π) de largeur au plus $b \in \mathbb{N}$, décide si $E(H) \setminus E(G)$ est (k, d, s, w) -arête colorable dans H . Nous appellerons cet algorithme **progDyn**.

Cependant, l'étape du choix des cartes planaires ayant un coût exponentiel en n , nous n'aurons pas obtenu d'algorithme FPT, paramétré par d et k , pour le problème plus général BFCP. Si donc nous nous intéressons au problème BCP, où le nombre total d'arêtes ajoutées est d'au plus k , la complexité du choix des cartes planaires dépend seulement de k . Cela va nous permettre d'obtenir un algorithme FPT, paramétré par d et k , pour le problème BCP. Cet algorithme fonctionne alors comme suit.

À partir d'un graphe plan quelconque, nous le transformons en un graphe lisse, puis nous rendons le graphe résultant 2-connexe via un choix de cartes planaires dans chaque face. Sur ce graphe, nous calculons une décomposition en coupes sphériques nous permettant d'appliquer la programmation dynamique décrite dans le chapitre précédent. Cette procédure est décrite par l'Algorithme 6. Notons que le Lemme 3 nous garantit la correction de cet algorithme.

Entrées : (G_{input}, d, k) une instance du problème BCP
Output : Une collection \mathcal{C} de (d, k) -configurations de X_e .

début
 si $\Delta(G_{\text{input}}) > k + 1$ alors
 | retourner Faux
 fin
 si $tw(G_{\text{input}}) > 3d + 1$ alors
 | retourner Faux
 fin
 $(G, J) = \text{grapheLisse}(G_{\text{input}})$
 pour $\bar{S} \in \mathbf{S}_G^{(k)}$ faire
 | si $\text{progDyn}(G, J, \tilde{G}_{\bar{S}}, d, k, 1, w_J)$ alors
 | | retourner Vrai
 | fin
 fin
 retourner Faux
fin

Algorithme 6: BCP

3.3.2 Complexité

Dans cette sous-section, nous allons donner la complexité de l'Algorithme 6 permettant de résoudre le problème BCP. Mais, nous devons tout d'abord donner la complexité d'une décomposition en coupes sphériques.

Lemme 6

Étant donné un graphe plan G de largeur arborescente au plus $b \in \mathbb{N}$. Nous pouvons alors construire une décomposition en coupes sphériques de G , dont la largeur est en $O(b)$, et ce en $n \cdot 2^{O(b)} + O(n^2)$ étapes.

Démonstration : Le Théorème 3, en remplaçant dans son énoncé q par b , nous permet de construire une décomposition arborescente, dont la largeur est en $O(b)$, et ce en $n \cdot 2^{O(d \cdot k)}$ étapes.

Ensuite, le Théorème 4, nous pouvons à partir de cette décomposition arborescente de G de largeur $O(b)$, construire une décomposition en branches de G de largeur $O(b)$ en $O(n^2)$ étapes.

Enfin, par le Théorème 6, nous savons qu'il est possible, à partir d'une décomposition en branches de G de largeur $O(b)$, de construire une décomposition en coupes sphériques de G d'une largeur de $O(b)$ en $O(n^2)$ étapes.

Ainsi, nous pouvons construire une décomposition en coupes sphériques de G de largeur $O(b)$ en $n \cdot 2^{O(b)} + O(n^2)$ étapes. ■

Enfin le théorème suivant donne le résultat central de ce mémoire.

Théorème 12

Étant donné un graphe plan G_{input} tel que $|V(G_{\text{input}})| = n \in \mathbb{N}$ et deux entiers $d, k \in \mathbb{N}$. Alors il existe un algorithme FPT, paramétré par d et k , et pouvant décider si (G_{input}, d, k) est une instance positive de BCP en $n^2 \cdot 2^{O(k)} + n \cdot 2^{O(d \cdot k)} + 2^{O(d \cdot k \cdot \log(d \cdot k))}$ étapes.

Démonstration : Nous savons par le Théorème 5 que $\mathbf{tw}(G_{\text{input}}) = O(d)$. Ensuite, le Lemme 1 nous assure que même en transformant G_{input} en un graphe lisse G , nous aurons $\mathbf{tw}(G) = O(d)$. Enfin, le Lemme 2 nous dit que quelque soit $\bar{S} \in \mathbf{S}_G^{(k)}$, nous aurons $\mathbf{tw}(\tilde{G}_{\bar{S}}) = O(k \cdot \mathbf{tw}(G)) = O(d \cdot k)$.

De plus, par le Lemme 1, nous savons que l'étape de transformation de G_{input} en le graphe lisse G s'effectue en $O(n)$ étapes.

Ensuite, et pour tout $\bar{S} \in \mathbf{S}_G^{(k)}$, nous calculons une décomposition en coupes sphériques pour le graphe $\tilde{G}_{\bar{S}}$ en $n \cdot 2^{O(d \cdot k)} + O(n^2)$ étapes. Et puisque nous nous intéressons au problème BCP, nous savons par le Corollaire 4 que $|\mathbf{S}_G^{(k)}| \leq 2^{O(k)}$.

Pour finir, et pour tout $\bar{S} \in \mathbf{S}_G^{(k)}$, nous appliquons la procédure **progDyn** sur l'instance $(G, J, \tilde{G}_{\bar{S}}, d, k, 1, w_J)$, où w est une fonction de poids sur les arêtes de G_{input} et J est l'ensemble d'arêtes de $E(G_{\text{input}})$ renvoyé par la procédure **grapheLisse** appliquée à G_{input} . Et, puisque nous nous intéressons au problème BCP, en remplaçant b par d dans le Corollaire 5, nous savons que cette procédure s'effectue en $2^{O(d \cdot k \cdot (\log(d \cdot k) + \log d))} = 2^{O(d \cdot k \cdot \log(d \cdot k))}$ étapes.

Tous ses arguments mis bout-à-bout nous permettent de conclure que la complexité finale de l'algorithme est en

$$\begin{aligned} & O(n) + 2^{O(k)} \cdot (n \cdot 2^{O(d \cdot k)} + O(n^2) + 2^{O(d \cdot k \cdot \log(d \cdot k))}) \\ &= 2^{O(k)} \cdot (n \cdot 2^{O(d \cdot k)} + O(n^2) + 2^{O(d \cdot k \cdot \log(d \cdot k))}) \\ &= n^2 \cdot 2^{O(k)} + n \cdot 2^{O(d \cdot k)} + 2^{O(d \cdot k \cdot \log(d \cdot k))}. \end{aligned}$$

■

Ce corollaire donne cette fois-ci la complexité de la procédure décrite par l'Algorithme 6 mais pour le problème BFCP et donc par extension pour le problème FBC.

Corollaire 7

Étant donné un graphe plan G_{input} tel que $|V(G_{\text{input}})| = n \in \mathbb{N}$ et deux entiers $d, k \in \mathbb{N}$. Alors il existe un algorithme pouvant décider si (G_{input}, d, k) est une instance positive de BFCP en $2^{O(n \cdot \log k)} \cdot (n \cdot 2^{O(d \cdot k)} + O(n^2) + 2^{O(d \cdot k \cdot \log(d \cdot k))})$ étapes.

CHAPITRE 4

Tableau récapitulatif et pistes de recherche

TABLE 4.1 – Résultats

	NP-COMPLET	FPT	ALGORITHME (temps d'exécution)
CPDB	Oui	Paramétré par d	Non
BCP	Oui	Paramétré par d et k	$n^2 \cdot 2^{O(k)} + n \cdot 2^{O(d \cdot k)} + 2^{O(d \cdot k \cdot \log(d \cdot k))}$
FBC	Oui	Non	$2^{O(n \cdot \log k)} \cdot (n \cdot 2^{O(d \cdot k)} + O(n^2) + 2^{O(d \cdot k \cdot \log(d \cdot k))})$
BFCP	Oui	Non	$2^{O(n \cdot \log k)} \cdot (n \cdot 2^{O(d \cdot k)} + O(n^2) + 2^{O(d \cdot k \cdot \log(d \cdot k))})$

Dans la Table 4.1, nous avons un tableau récapitulatif des différents résultats obtenus par nous et par d'autres sur les problèmes autour de la COMPLÉTION PLANE À DIAMÈTRE BORNÉ.

Nous proposons dans ce chapitre, et avant de conclure, une série d'améliorations possibles de ces résultats, ainsi que certaines pistes pour des recherches futures.

4.1 Améliorations possibles

4.1.1 Connecter le graphe

En premier, il faudrait arriver à trouver une autre technique que les choix de cartes planaires pour connecter le graphe. Une proposition possible est de placer dans chaque face dont la frontière est union d'au moins deux cycles une grille annulaire. Les cycles de cette face seraient bien sûr augmentés d'une grille annulaire comme dans la version proposée dans ce mémoire. Cette grille supplémentaire serait connectée par un "pont" composé de k arêtes parallèles, i.e. formant deux-à-deux une face de taille quatre, avec chacun des cycles.

Nous pensons disposer d'une preuve de la capacité d'une telle construction à simuler toutes les connections possibles, de plus, il nous apparaît que sa largeur arborescente n'augmenterait pas d'un facteur de plus de k , exactement comme la construction proposée dans ce mémoire. Mais prouver ce dernier point semble assez compliqué et nécessiterait certainement des techniques avancées de calcul de la largeur arborescente.

Dans ce cas, la construction dans chaque face est immédiate et nous voilà débarrassé de la complexité si gênante du choix des cartes planaires. Notre programmation dynamique permettrait ainsi de prouver que les deux problèmes FBC et surtout BFCP sont FPT paramétrés par d et k . La complexité de l'algorithme en l'état serait alors la suivante :

$$(n \cdot 2^{O(d \cdot k)} + O(n^2) + 2^{O(d \cdot k \cdot \log(d \cdot k))}).$$

4.1.2 La fonction γ

Deuxièmement, il serait sans doute possible, étant donné que le graphe est plan, de trouver une structure de Catalan modélisant toutes les manières possibles pour deux sommets de l'intérieur d'un lasso d'être potentiellement proches. Trouver une structure en ce sens c'est en quelque sorte considérer le paramètre d , inhérent au problème, comme un paramètre structurel du graphe. Cette structure de Catalan permettrait de réduire la taille du stockage de la fonction γ en une fonction simple exponentielle en d et en k .

De plus, si la taille de la fonction γ devient simple exponentielle en d et en k , le temps de calcul de la fonction γ pourrait aussi être réduit via une structure de Catalan. Et pourrait donc devenir simple exponentiel en d et en k . Il nous semble qu'une telle amélioration est faisable. Implémenter cette structure serait un plus pour la complexité finale, qui deviendrait ainsi

- en $n^2 \cdot 2^{O(k)} + n \cdot 2^{O(d \cdot k)} + 2^{O(d \cdot k \cdot (\log d + \log k))}$ pour BCP et
- en $2^{O(n \cdot \log k)} \cdot (n \cdot 2^{O(d \cdot k)} + O(n^2) + 2^{O(d \cdot k \cdot (\log d + \log k))})$ pour FBC et BFCP.

Notons que la complexité du calcul du graphe G serait bien entendu réduite par la même occasion.

4.2 Pistes de recherche

4.2.1 Grille annulaire et problèmes de complétion

De nombreuses pistes de recherche s'offrent à nous. Tout d'abord, l'objectif en arrière plan de ce mémoire est de systématiser l'étude de la complexité paramétrée des problèmes de complétion d'arêtes à partir de la complexité paramétrée de leur problème originel correspondant, i.e. dans notre cas le problème de mesurer le diamètre d'un graphe est polynomial en la taille de la donnée. En particulier, si le problème originel est FPT paramétré par le problème lui-même, par exemple la taille du mineur pour le problème de vérification de mineur, alors le problème de complétion l'est-il aussi ?

Notons qu'un résultat négatif de ce genre a déjà été montré lorsque le paramètre est le nombre d'arêtes à ajouter. Mais que dire lorsque le paramètre dépend du problème ou encore est un paramètre structurel du graphe, comme la largeur arborescente ?

Notre méthode de simulation d'ajout d'arêtes par une structure radiale pourrait sans doute être utilisée dans d'autres problèmes de complétion plane. En particulier, nous fondons des espoirs dans son adaptation au problème suivant :

COMPLÉTION PLANE EN ISOMORPHISME DE SOUS-GRAPHE

Entrée : deux graphes plan G et H .

Question : existe-t-il une complétion plane G^+ de G admettant un sous-graphe isomorphe à H ?

Ainsi qu'aux problèmes d'échanges d'arêtes, dits "flips" en anglais, dans des triangulations soit de graphes plans, soit de polygones.

De même, le problème de l'arête-colorabilité nécessiterait d'être clarifié et peut-être même réduit polynomialement à un problème NP-COMPLET. Dans ce cas, il pourrait très certainement être adaptable à d'autres problèmes proches.

La technique de la grille annulaire nécessite cependant l'existence de faces et ne peut donc être généralisée aux graphes planaires, si ce n'est les graphes planaires 3-connexes.

Cependant, nous pourrions aussi chercher à adapter notre programmation dynamique pour couvrir le cas orienté.

4.2.2 Problèmes ouverts

Deux problèmes ouverts nous semblent intéressants. Tout d'abord, celui déjà proposé par Fellows et Diejter qui reste donc jusqu'à maintenant toujours ouvert.

Problème ouvert 1 : trouver un algorithme réalisable et FPT paramétré par d pour le problème CPDB.

Et enfin celui-ci que nous proposons et qui généralise le problème du stockage de la fonction γ .

Problème ouvert 2 : étant donné un disque borné du plan et un graphe planaire plongé sans croisement dans l'intérieur du disque tel que le bord du disque admette un nombre b de sommets. Combien y a-t-il de chemins, de longueur bornée par une constante c , différents en terme de taille, entre un sommet intérieur du graphe et les sommets du bord du disque ?

Conclusion

Nous avons en premier lieu défini le cadre des problèmes de complétion plane, ainsi que celui de leur complexité paramétrée.

Nous avons ensuite défini les notions de décomposition de graphes, en particulier la décomposition arborescente et la décomposition en branches ainsi que leurs notions rattachées que sont la largeur arborescente et la largeur de branche.

Cela, nous a donné la possibilité d'introduire différentes techniques permettant de construire des algorithmes FPT, en particulier la programmation dynamique sur une décomposition en coupes sphériques d'un graphe.

Ensuite, nous nous sommes intéressés au problème de la COMPLÉTION PLANE À DIAMÈTRE BORNÉ paramétré par le diamètre d .

Et nous avons défini trois variantes de ce problème paramétré l'une par le nombre total d'arêtes à ajouter, l'autre par la taille maximum d'une face du graphe de départ et enfin par le nombre maximum d'arêtes à ajouter par face.

Cette dernière étant la plus générale, nous avons montré qu'elle était équivalente au problème de la (d, k, s, w) -arête colorabilité. Nous avons alors proposé une transformation de graphe conservant dans une certaine mesure les paramètres structuraux du graphe de départ.

Et sur cette transformation, nous avons développé une programmation dynamique sur une décomposition en coupes sphériques.

De cette programmation dynamique, nous en avons tiré un algorithme qui s'est avéré être FPT paramétré par d et le nombre total d'arêtes à ajouter, i.e. pour la première variante, mais pas pour les deux autres.

Nous avons enfin proposé un certain nombre d'améliorations possibles de notre algorithme ainsi que des pistes de recherche futures afin d'utiliser les techniques développées dans ce mémoire pour d'autres problèmes.

Bibliographie

- [Albenque and Poulalhon, 2013] Albenque, M. and Poulalhon, D. (2013). Generic method for bijections between blossoming trees and planar maps. arXiv :1305.1312.
- [Arnborg et al., 1987] Arnborg, S., Corneil, D. G., and Proskurowski, A. (1987). Complexity of finding embeddings in a k-tree. *SIAM J. Algebraic Discrete Methods*, 8(2) :277–284.
- [Arnborg et al., 1990] Arnborg, S., Proskurowski, A., and Corneil, D. G. (1990). Forbidden minors characterization of partial 3-trees. *Discrete Mathematics*, 80(1) :1 – 19.
- [Baker, 1994] Baker, B. S. (1994). Approximation algorithms for np-complete problems on planar graphs. *J. ACM*, 41(1) :153–180.
- [Bodlaender, 1988] Bodlaender, H. L. (1988). Some classes of graphs with bounded treewidth. *Bulletin of the EATCS*, 36 :116–125.
- [Bodlaender, 1997] Bodlaender, H. L. (1997). Treewidth : Algorithmic techniques and results. In Prívvara, I. and Ružička, P., editors, *Mathematical Foundations of Computer Science 1997*, volume 1295 of *Lecture Notes in Computer Science*, pages 19–36. Springer Berlin Heidelberg.
- [Bodlaender, 1998] Bodlaender, H. L. (1998). A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1–2) :1 – 45.
- [Bodlaender, 2005] Bodlaender, H. L. (2005). Discovering treewidth. In Vojtáš, P., Bieliková, M., Charron-Bost, B., and Sýkora, O., editors, *SOFSEM 2005 : Theory and Practice of Computer Science*, volume 3381 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin Heidelberg.
- [Bodlaender et al., 2013] Bodlaender, H. L., Drange, P. G., Dregi, M. S., Fomin, F. V., Lokshtanov, D., and Pilipczuk, M. (2013). An $O(c^k n)$ 5-approximation algorithm for treewidth. In *FOCS*, pages 499–508. IEEE Computer Society.
- [Cai, 1996] Cai, L. (1996). Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4) :171 – 176.
- [Chung, 1987] Chung, F. R. K. (1987). Diameters of graphs : Old problems and new results. *Congressus Numerantium*, 60.

- [Dejter and Fellows, 1993] Dejter, I. J. and Fellows, M. R. (1993). Improving the diameter of a planar graph. manuscript.
- [Diestel, 2012] Diestel, R. (2012). *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer.
- [Downey and Fellows, 1999] Downey, R. G. and Fellows, M. R. (1999). *Parameterized Complexity*. Springer-Verlag, New York.
- [Eppstein, 1995] Eppstein, D. (1995). Subgraph isomorphism in planar graphs and related problems. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '95, pages 632–640, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.
- [Eppstein, 2000] Eppstein, D. (2000). Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27 :275–291. Special issue on treewidth, graph minors, and algorithms.
- [Eppstein, 2010] Eppstein, D. (2010). File :branch-decomposition.svg.
- [Flum and Grohe, 2006] Flum, J. and Grohe, M. (2006). *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [Garey and Johnson, 1979] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.
- [Koutsonas and Thilikos, 2011] Koutsonas, A. and Thilikos, D. M. (2011). Planar feedback vertex set and face cover : Combinatorial bounds and subexponential algorithms. pages 987–1003.
- [Kreweras, 1972] Kreweras, G. (1972). Sur les partitions non croisees d'un cycle. *Discrete Mathematics*, 1(4) :333–350.
- [Mohar and Thomassen, 2001] Mohar, B. and Thomassen, C. (2001). *Graphs on Surfaces*. Graphs on Surfaces. Johns Hopkins University Press.
- [Robertson and Seymour, 1986] Robertson, N. and Seymour, P. D. (1986). Graph minors. ii. algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3) :309 – 322.
- [Robertson and Seymour, 1991] Robertson, N. and Seymour, P. D. (1991). Graph minors. x. obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2) :153 – 190.
- [Seymour and Thomas, 1994] Seymour, P. D. and Thomas, R. (1994). Call routing and the ratcatcher. *Combinatorica*, 14(2) :217–241.
- [Sharan, 2002] Sharan, R. (2002). *Graph Modification Problems and their Applications to Genomic Research*. PhD thesis, Tel-Aviv University.
- [Vazirani, 2001] Vazirani, V. V. (2001). *Approximation Algorithms*. Springer-Verlag New York, Inc., New York, NY, USA.
- [Yannakakis, 1981] Yannakakis, M. (1981). Computing the minimum fill-in is np-complete. *SIAM Journal on Algebraic Discrete Methods*, 2(1) :77–79.